BIOINFORMATICS

Vol. 20 no. 18 2004, pages 3628–3635
doi:10.1093/bioinformatics/bth460

Modular, scriptable and automated analysis tools for high-throughput peptide mass fingerprinting

Jim Samuelsson[1], Daniel Dalevi[2], Fredrik Levander[3] and Thorsteinn Rögnvaldsson[4,*]

[1]Genedata GmbH, Lena-Christ-Strasse 50, 82152 Martinsried, Germany, [2]Computing Science, Chalmers University of Technology, SE-412 96 Göteborg, Sweden, [3]Department of Protein Technology, Lund University, Sölvegatan 33A, SE-223 62 Lund, Sweden and [4]School of Information Science, Computer and Electronic Engineering, Halmstad University, Box 823, SE-301 18 Halmstad, Sweden

Received on February 20, 2004; revised on May 13, 2004; accepted on July 27, 2004.
Advance Access publication August 5, 2004

ABSTRACT

Summary: A set of new algorithms and software tools for automatic protein identification using peptide mass fingerprinting is presented. The software is automatic, fast and modular to suit different laboratory needs, and it can be operated either via a Java user interface or called from within scripts. The software modules do peak extraction, peak filtering and protein database matching, and communicate via XML. Individual modules can therefore easily be replaced with other software if desired, and all intermediate results are available to the user. The algorithms are designed to operate without human intervention and contain several novel approaches. The performance and capabilities of the software is illustrated on spectra from different mass spectrometer manufacturers, and the factors influencing successful identification are discussed and quantified.

Motivation: Protein identification with mass spectrometric methods is a key step in modern proteomics studies. Some tools are available today for doing different steps in the analysis. Only a few commercial systems integrate all the steps in the analysis, often for only one vendor's hardware, and the details of these systems are not public.

Results: A complete system for doing protein identification with peptide mass fingerprints is presented, including everything from peak picking to matching the database protein. The details of the different algorithms are disclosed so that academic researchers can have full control of their tools.

Availability: The described software tools are available from the Halmstad University website www.hh.se/staff/bioinf/

Supplementary information: Details of the algorithms are described in supporting information available from the Halmstad University website www.hh.se/staff/bioinf/

1 INTRODUCTION

Mass spectrometry (MS) has become a standard tool for the identification of proteins and mapping of proteomes, using various technologies: matrix-assisted laser desorption/ionization (MALDI), electrospray ionization (ESI), ion traps, quadrupole and time-of-flight (TOF) spectrometers. For high-throughput protein identification, the most common procedure today is two-dimensional (2D) gel separation followed by enzymatic digestion, MALDI-TOF mass measurement and a peptide mass fingerprint (PMF). Identifying a protein using PMF is a multistep process; it requires extracting monoisotopic peaks from a mass spectrum, calibrating the spectrum, removing contaminant peptide peaks and matching the resulting list of monoisotopic peaks with expected theoretical peptide monoisotopic masses. The quality of each individual step affects the sensitivity and reliability of the final protein identification, and knowledge about what data processing was done, or will be done, in a previous/later step can be used to improve the data processing in one step of the process. For instance, the monoisotopic peak picking step can be done for a range of parameter settings to increase the sensitivity of the process (Rögnvaldsson *et al*., 2004), and knowledge about the parameter settings for the database match can be used to calibrate the spectrum and remove contaminant peaks (Gobom *et al*., 2002). However, common software systems do not allow the user the control to iterate between steps or the freedom to combine different favorite tools or algorithms for each step.

A suite of software modules and algorithms that perform the different steps of PMF is presented in this paper. The software, mostly written in C++, builds on several original ideas, is fast, flexible and works under different conditions and platforms (e.g. Windows, Unix, Linux, etc.). It can be used in a streamlined fashion where each module feeds its results onward to the next module, for high-throughput PMF, or in a user-interactive setting with a graphical user interface (GUI).

*To whom correspondence should be addressed.

It can also be called from, e.g. a web interface or from scripts within other programs. The different modules take their input and produce their output in XML.

The description of the different modules is kept brief and a full account of the algorithms is provided in the Supplementary information.

2 METHODS

The software consists of three key modules: the peak extraction module, the peak post-processing module and the peptide fingerprinting module.

2.1 Peak extraction

The peak extraction is done in four steps: estimate baseline and noise level, construct peaks, cluster the peaks and deisotope the clusters. In the discussion below, x and y denote the values of m/z (mass/electric charge) and intensity, respectively.

2.1.1 Baseline and noise level estimation Baseline and noise levels are estimated with a weighted average of the minimum and maximum peak values, respectively, within windows of small mass intervals. The weighting downweights regions with large peaks in relation to regions with small peaks. The noise level therefore equals the height of the smallest peaks at a given resolution (a few Dalton) and the baseline equals the minimum intensities at the same resolution.

2.1.2 Peak construction and clustering All datapoints whose y values are above the estimated noise level ($s/n = 1$) are used to construct peaks; a peak is defined as a consecutive sequence of datapoints with y values above $s/n = 1$. The location of a peak in the x direction is determined by a centroid calculation, and the intensity is taken as the maximum y value in the peak.

All constructed peaks are then grouped into clusters, where a peak cluster consists of peaks distanced 1 ± 0.2 Da apart; a cluster must contain at least one peak whose signal-to-noise level is above the threshold trigger level set by the user (or else it is discarded).

2.1.3 Monoisotope identification Each peak in a peak cluster with R peaks can, in theory, be a monoisotopic peak for a peptide. Each monoisotope is accompanied by a number of isotopic peaks with known relative intensities (relative w.r.t. the monoisotopic peak intensity). This can be expressed as $\mathbf{y} \approx \mathbf{Ma}$, with the intensity column vector $\mathbf{y} = (y_1, y_2, \ldots, y_R)^{\mathrm{T}}$, 'abundancy' column vector $\mathbf{a} = (a_1, a_2, \ldots, a_R)^{\mathrm{T}}$, and isotopic $R \times R$ distribution matrix $\mathbf{M}$. The matrix $\mathbf{M}$ contains the expected isotopic relative intensities at the peak cluster mass, and the 'abundancy' element a_k is the (non-negative) contribution to the peak cluster from a monoisotope located at peak k in the cluster. The deisotoping problem is therefore formulated as

$$\text{minimize } H = (\mathbf{Ma} - \mathbf{y})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{Ma} - \mathbf{y}) \qquad (1)$$

subject to the constraints that all components of the vector $\mathbf{a}$ are non-negative. Here, $\mathbf{M}$ is a matrix with the template isotope and $\boldsymbol{\Sigma}$ is a diagonal matrix with the peak intensity uncertainties. Equation (1) is a quadratic programming problem that is solved reliably and fast using standard algorithms; Schittkowski (2003, http://www.uni-bayreuth.de/departments/math/~kschittkowski/) provides an example of a robust and quick algorithm for this. The minimization procedure corresponds to the objective of determining the lowest number of peptides, and their m/z values, which, given the measured peak intensities and the template isotope distributions, can account for the isotopic pattern of the cluster.

The output from the peak extraction module is an XML formatted list of likely monoisotopic peptides and their properties, e.g. mass, intensity, signal-to-noise ratio, peak width and ion current.

2.1.4 Peak extraction discussion The quadratic programming deisotoping is novel and different from previously published methods (Gras *et al*., 1999; Field *et al*., 2002; Berndt *et al*., 1999; Breen *et al*., 2000). It allows fitting for several overlapping peptides at once, in contrast to previous approaches that use an iterative procedure where the contribution from each peptide is subtracted from the spectrum and a new fit is done in order to find overlapping peptides. Furthermore, our approach allows for including a prior into the cost, which expresses the prior probability for finding a certain number of peptides with approximately the same mass (this is shown in the supporting information).

The deisotoping step allows custom isotopic distributions; the template isotopic distribution is provided in a file and users can specify their own preferences.

2.2 Peak list post-processing

Submitting the peak masses from the peak extraction module directly to a peptide fingerprinting algorithm is usually a bad idea because the list can be poorly calibrated and also contain contamination peak masses. The peak list must therefore be post-processed before it is submitted to the PMF module. This post-processing is done in four main steps: removal of spurious peaks, internal calibration, removal of too low and too high masses, and removal of unwanted contamination peaks.

2.2.1 Removing spurious peaks The spectrum will sometimes contain spurious peaks, e.g. a peak 'echo' following a large peak or peaks that result from small spikes in the spectrum. Small peaks that lie very close to large peaks and peaks that are very narrow are therefore by default removed (these peaks can be kept if the user wishes so).

2.2.2 Internal calibration The peak list is compared to a list of expected reference masses, which can be protease autolysis peptides, human keratin peptides and/or matrix peaks. These reference peaks are provided in a filter parameter file, which is either created by the user or automatically generated using a batch of spectra; the latter is described in detail in a separate paper (Levander *et al*., 2004). The masses in the peak list that match reference peaks are used to determine the parameters (a, b, c) in a robust regression (outliers are detected and removed) of the form

$$(m/z)_{\text{new}}^{1/2} = a(m/z)_{\text{old}} + b(m/z)_{\text{old}}^{1/2} + c, \tag{2}$$

where $(m/z)_{\text{new}}$ is the mass value after calibration and $(m/z)_{\text{old}}$ is the mass value before calibration. The form of this equation was chosen by experiment, but similar approaches have been presented before.

Automatic calibration can go wrong and the calibrated masses, $(m/z)_{\text{new}}$, are therefore always compared against allowed biological peptide masses (Mann, 1995; Gay *et al*., 1999) before the calibration is accepted. If the calibration yields too large changes, as specified by the user, or if the likelihood for the calibrated monoisotopic peak masses is too low, given the distribution of allowed biological peptide masses, then the calibration is discarded.

2.2.3 Removing low and high masses Every MS experiment has a mass range within which the data are considered to be of high quality. It is common practice to discard low mass peaks because of the matrix contamination. Furthermore, the experimental sensitivity decreases towards high masses and it is important for the peptide fingerprint matching to correctly specify the maximum expected mass. All masses that fall outside of user-specified minimum and maximum masses are therefore removed from the peak list. This information is also passed onto the PMF module (described below) since it needs to know the mass range within which peptides can be detected in order to compute a precise match score.

2.2.4 Removing other unwanted masses Every spectrum contains a number of peaks that originate from, e.g. protease autolysis and human skin proteins (Parker *et al*., 1998; Karty *et al*., 2002). Purging these from the peak list improves the statistics in the peptide fingerprint match (provided that the unknown proteins in the sample do not have many peptides that overlap with contamination peaks). The final step in the post-processing is therefore to remove such contamination masses. The post-processing yields a new peak list, $\mathbf{x} = (x_1, x_2, \ldots, x_L)$, where a larger fraction of the peaks come from the interesting proteins, and the mass precision is significantly improved.

2.2.5 Peak post-processing discussion The peak list post-processing is probably the single most important step in the PMF analysis chain. Our own experiments indicate that post-processing can double the success rate in a PMF experiment (Levander *et al*., 2004), compared to using the raw data coming out of the MS. Chamrad *et al*. (2003) report a 7-fold increase. It is, however, important to note that proper post-processing means removing essentially all the contaminant peaks, calibrating each spectrum with several reference masses (not just two or three) and providing correct information on the valid mass range to the PMF database matching module. The proper way to find all contaminant peaks in a batch of spectra is to compare them all and identify peaks that occur unreasonably often, as described by Levander *et al*. (2004). Other suggested approaches (Chamrad *et al*., 2003; Hjernø *et al*., 2002) use a predefined cut-off, e.g. peaks occuring in 15% of the spectra, which is statistically incorrect since it ignores the varying probability for observing peptides in different mass ranges.

2.3 Peptide mass fingerprinting

The final step in the PMF analysis chain is the peptide mass fingerprinting itself. This can, given an experimental peak list **x** and a protein database, be divided into three separate steps: creating theoretical spectra for the database entries, computing the match between the empirical spectrum and the theoretical spectra, and assessing the matching scores.

2.3.1 Creating theoretical spectra The enzymatic digestion that was carried out in the laboratory is mimicked in the computer for each entry in the database, also called a database protein. The chemical rules for digestion, possible post-translational modifications and possible missed cleavages are applied, leading to a list of expected peptide masses from each database protein. We denote the j-th entry in the protein database by $T[j]$ and the corresponding expected set of peptide masses by $\mathbf{z}(j) = [z_1(j), z_2(j), \ldots, z_{N(j)}(j)]$, where $N(j)$ is the number of peptides that resulted when $T[j]$ was digested *in silico*.

2.3.2 Computing the spectrum match There are many ways to express the match between an experimental peak list **x** and a theoretical spectrum $\mathbf{z}(j)$ (Pappin *et al*., 1993; Zhang and Chait, 2000; Clauser *et al*., 1999; Perkins *et al*., 1999; Wool and Smilansky, 2002). Our approach is to consider strong spectrum resemblance an unlikely event, which is also the approach taken in the ProtoCall tool (Wool and Smilansky, 2002) and MASCOT (Perkins *et al*., 1999) (it is unknown to us exactly how the MASCOT score is computed). This is reasonable because there is only one correct database protein (or a small group of correct database proteins), if any, and we expect only the correct protein(s) to show strong spectrum resemblance. Non-correct proteins are not expected to show strong spectrum resemblance; if they did, MS would not be a suitable tool for protein identification. The algorithm therefore computes, for each database protein $T[j]$, the a priori random probability for the set of peaks shared between $\mathbf{z}(j)$ and **x** to occur. Proteins where the observed match is unlikely to occur by chance are considered better candidates than proteins where the match is more likely.

There are different ways to compute such a probability, basically reflecting the level of approximation one is willing to accept. Two different expressions for this probability are derived in the Supplementary information; only the expression used in the default version of the software (case 1 in the Supplementary information) is described here. The a priori probability that $\mathbf{x}$ and $\mathbf{z}(j)$ have r common peaks within a tolerance window δ is

$$p(j) \equiv p(r \mid \mathbf{z}(j), \mathbf{x}, \delta) = \binom{L}{r} P^r Q^{L-r}, \tag{3}$$

where L is the number of peaks in the experimental peak list $\mathbf{x}$, $P = P(\delta)$ is the probability for at least one match between a peak from the experimental peak list $\mathbf{x}$ and one of the $N(j)$ peptide masses of $T[j]$, and $Q = 1 - P$. A similar expression is used in the ProtoCall search tool (Wool and Smilansky, 2002).

Protein $T[j]$ is given a score value by taking the negative natural logarithm of the probability

$$\sigma(j) \equiv \sigma[\mathbf{z}(j), \mathbf{x}, \delta] \equiv -\ln[p(j)]. \tag{4}$$

All database proteins are thus given a score value according to Equation (4) and if $\sigma(j) > \sigma(i)$ then $T[j]$ is considered to be a more likely candidate protein than $T[i]$.

2.3.3 Assessing the score values Obviously, there is always one protein from the protein database with the highest score value, irrespective of whether the unknown protein is registered in the database as an entry or not. It is therefore necessary to asses the scores (Eriksson *et al.*, 2000; Fenyö and Beavis, 2003; Berndt *et al.*, 1999).

The assessment is such that we want to make sure that the best scoring database proteins indeed have score values that are significantly higher than what one would expect from just random trials. By a random trial, we mean a comparison between the experimental peak list $\mathbf{x}$ and a theoretical peak list $\mathbf{z} = (z_1, \ldots, z_N)$ whose peak values have been selected from a random distribution. This random distribution is generated at the same time as the proteins in the database are digested; the assumption being that the distribution of peptides from enzyme digests is fairly uniform over all proteins and that the current database does not constitute a non-typical representation of all proteins. That such an assumption is reasonable gets support from the fact that distributions of tryptic peptide masses for different genomes show high similarity (Fenyö *et al.*, 1998).

Given the experimental peak list $\mathbf{x}$ and the set of peak match tolerance windows $\boldsymbol{\delta}$ (one or many) the probability to get, during random trials, a score value higher than σ_c is

$$\begin{aligned} P_{\mathrm{rnd}}(\sigma_c) &\equiv P_{\mathrm{rnd}}[\sigma > \sigma_c \mid (\mathbf{x}, \delta)] \\ &= \int \mathrm{d}\mathbf{z}\psi(\mathbf{z})\Theta[\sigma_c - \sigma(\mathbf{z}, \mathbf{x}, \delta)], \end{aligned} \tag{5}$$

where $\psi(\mathbf{z})$ is the probability density for peak lists of digested proteins (described by $\mathbf{z}$) and Θ is the Heaviside step function [$\Theta(t) = 1$ if $t > 0$, $\Theta(t) = 0$ otherwise]. The integral is estimated over all possible vectors $\mathbf{z}$ using Monte Carlo integration. With P_{rnd} it is straightforward to derive two assessment measures that are intuitively easy to understand, the well-known P-value and, also, another measure called the quality, $\mathcal{Q}$.

The P-value expresses the probability for getting the observed result if the null hypothesis is true. Our null hypothesis is that $\mathbf{x}$ matches a protein with random peptide masses better than it matches database protein $\mathbf{z}(j)$, so the P-value for score σ_j is

$$P\text{-value } = 1 - [1 - P_{\mathrm{rnd}}(\sigma > \sigma_c)]^D, \tag{6}$$

where D is the size of the database. If $P_{\mathrm{rnd}} \ll 1$, then we have p-value $\approx D \cdot P_{\mathrm{rnd}}$. It is customary to reject the null hypothesis when the P-value is <5%.

The mathematical derivation of quality is given in the Supplementary information; here, we describe the way it should be interpreted in a database search. If

- the search is done in a protein database where the number of protein entries is D (typical values of D are today 10^5–10^6);
- the highest score value for all database proteins is σ_{top};
- the quality value for σ_{top} is $\mathcal{Q}(\sigma_{\mathrm{top}}) \equiv \mathcal{Q}_{\mathrm{top}}$.

then one would need to make $D \exp(\mathcal{Q}_{\mathrm{top}})$ random trials in order to expect a score value of σ_{top}. Another, intuitively appealing, way of expressing this is that one would need a random protein database the size of $D \exp(\mathcal{Q}_{\mathrm{top}})$ in order to expect to observe a score value of σ_{top} the same number of times as was done in the database with real proteins. Therefore, if $\mathcal{Q}_{\mathrm{top}} \approx 0$ then one would not really trust the top candidate to be the unknown protein. On the other hand if, for example, $\mathcal{Q}_{\mathrm{top}} > 7$ then one would need a random database at least 1000 times larger than the real database in order to observe a score value of σ_{top}.

Significance of a search result is often presented in terms of an E (expectation)-value in sequence alignment searches, cf. BLAST (Altschul *et al.*, 1997). A corresponding E-value for a score σ is

$$E(\sigma) = n(\sigma) \cdot \exp[-\mathcal{Q}(\sigma)], \tag{7}$$

where $n(\sigma)$ is the number of database proteins with a score value of at least σ.

2.3.4 PMF database matching discussion The PMF database matching is a pattern recognition problem; we want to determine the most probable protein(s) in the mixture, given the measured mass spectrum. In theory, to do this in an optimal way we should estimate the a posteriori probability $p(T[j] \mid \mathbf{x})$ for all proteins $T[j]$ in the database. Doing this properly requires a good model for the probability of observing a specific peptide, the prior probability $p(T[j])$

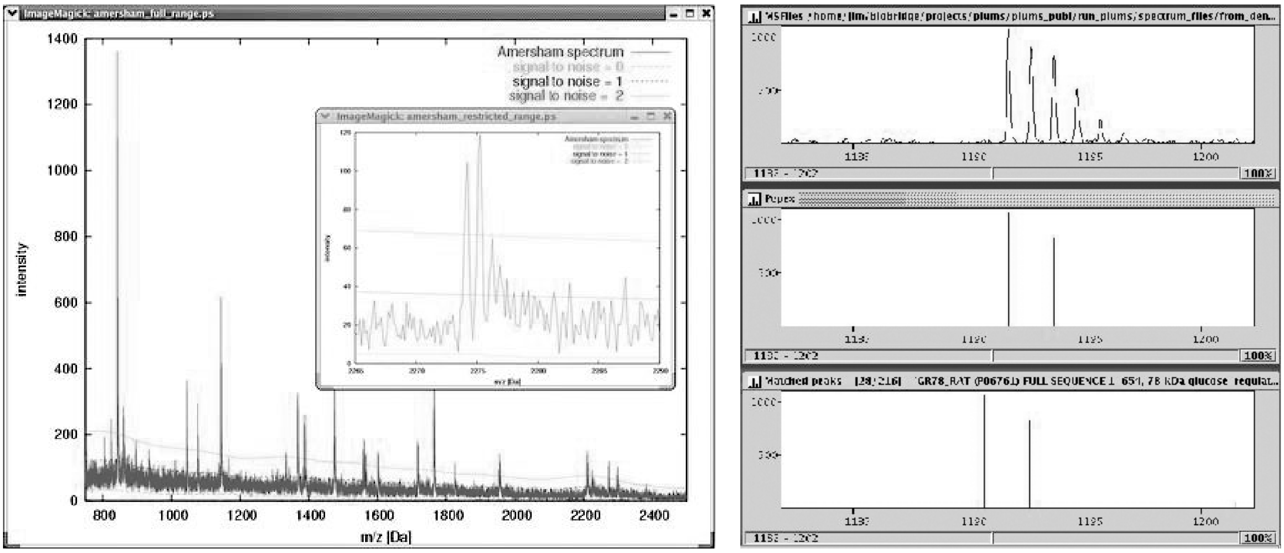

Fig. 1. Left panel: A spectrum where the curves for the levels of $s/n = 0$, $s/n = 1$ and $s/n = 2$ (the user-specified threshold) are shown. All datapoints above the $s/n = 1$ level are used to build up the identified peaks. A peak cluster, typical for the higher mass range, is shown in the inset. Right panel: A screen dump from the GUI showing a peak cluster where the peak extraction selects the first and the third peaks of the cluster; both peaks survive the filtering step and both match the top scoring protein. The two top panels show the M+H ion masses whereas the bottom panel shows the peptide masses without the extra proton.

for observing a certain protein, and the mass error in the MS, which becomes a very complicated issue.

Most probability based algorithms (Perkins *et al.*, 1999; Wool and Smilansky, 2002), including ours, instead try to answer a simpler but related question: what is the probability that the observed peptide masses in the spectrum matches a random sequence of peptides? If this probability is very low then perhaps the best matching protein is indeed in the sample, which often is the case. The procedure is the same as in a standard hypothesis test; the null hypothesis being that the observed peptide list is random and the null hypothesis is rejected if the probability for the observed score is very low.

The expression in Equation (3) is based on assuming that it is equally likely to observe any expected peptide from the database protein. We have, for example, refrained from assuming that arginine-terminated peptides should be more prominent than lysine-terminated peptides. Other scores have been suggested (Parker, 2002; Magnin *et al*., 2004) that make more specific use of the amino acid composition in the peptides and it is straightforward to modify expression (3) if certain peptides are not expected (just decrease the value of L). However, introducing such a probability is guesswork, unless one has access to MS data of an amount large enough for making statistically reliable estimates, and specific enough for being relevant to the experimental condition under investigation. It is also uncertain how well such an algorithm would generalize across different laboratories where slightly different experimental protocols are used. Equation (3) therefore assumes nothing about the specific features of the experiment studied.

3 RESULTS AND DISCUSSION

3.1 Performance on sample spectra

The software tools are demonstrated using spectra from three different manufacturers: MALDI-TOF MS from Amersham Biosciences (Ettan™), Applied Biosystems (Voyager-DE™) and Bruker Daltonics (autoflex®). The first two spectra are shown in Figure 1, and the third is shown in the Supplementary information. The second and third spectra come from batches of spectra from samples on the same gel where the software automatically identified calibration and contamination peaks, which enables comparison of the human operator and the software. The fixed settings, for all spectra and all searches, were as follows: $s/n = 2.0$; peak matching window: 0.1 Da; database: Swissprot40; digestion by trypsin; chemical modifications are carbamidmethylation of cysteine (fixed) and oxidation of methionine (variable), and allowing for one missed cleavage. The human operator filters contained known autolysis peaks from trypsin (bovine for the Applied spectrum, porcine for the two others) and hair-α keratin. The identities of the top scoring proteins for each spectrum and their quality values Q are shown in Table 1.

The left panel of Figure 1 shows the estimated baseline, noise and peaks in the peak extraction step. The upper dashed line is the signal-to-noise level set by the user ($s/n = 2$ in this example), the middle dashed line is the $s/n = 1$ level and the

Table 1. Identification results when using spectra from different manufacturers

Filter and calibration	Amersham Top ID	Q	Applied Top ID	Q	Bruker Top ID	Q
No	DLDH_ECOLI	11.48	GR78_RAT	7.38	DHE3_RAT	2.36
Human	DLDH_ECOLI	12.72	GR78_RAT	7.90	DHE3_RAT	2.68
Batch	—	—	GR78_RAT	9.97	ATPA_HUMAN	11.97
	—	—			DHE3_HUMAN	4.84

A dash (—) means that the setup was not tried.

lower dashed line is the $s/n = 0$ level. All datapoints between the two lower lines constitute the noise (chemical, electronic, etc.). The datapoints above the $s/n = 1$ level are used to build up the peaks. The right panel of Figure 1 shows a peak cluster from the Applied spectrum, at ~1200 Da. It consists of five peaks and the peak extraction selects the first and the third peaks (counting from the left) as monoisotopes. The isotopic pattern cannot be explained by one single peptide but two peptides are sufficent. Both peaks survive the post-processing step and both match peak masses of the top scoring protein: GR78_RAT.

Given the robust peak extraction, the identification is rather undramatic for the Applied and Amersham spectra, with high statistical significance (high Q-values) both with and without filters. For the Bruker spectrum, which comes from a human sample, the situation is different. The software does not report any significant candidates for the first two filter alternatives, no filter and human operator. It is not until the automatically generated filter is applied that significant candidates are reported; the filter was generated using the algorithm described by Levander *et al*. (2004). Besides the extremely confident identification of ATPA_HUMAN, there is a likely identification (defined as $Q > 3$) for DHE3_HUMAN. This indicates that the sample contains a mixture of these two proteins, which is possible since the two proteins are reasonably close in isoelectric point and molecular weight. The reason for the dramatic increase in score quality is that the automatically generated filter has identified several trypsin autodigestion peaks that are non-regular, i.e. not adjacent to lysine or arginine. Such effects are hard for a human operator to detect, or foresee, but straightforward for the software (more extended examples of the role of filters are provided in the Supplementary information).

3.2 Comparison to other PMF tools

All steps in a PMF experiment are important and influence each other, why information should be carried between steps. Furthermore, it should be simple to combine individual steps with customized tools and/or alternative favorite tools, why intermediate results should be easily available (e.g. if one wants to apply one's own calibration routine). This flexibility is, however, not offered by most available tools for PMF analysis.

Table 2. Performance of our PMF tool (Piums), compared with Mascot and ProFound, on 266 spectra with yeast protein samples

	Piums (v. 3.0.10)	Mascot (v. 1.9)	ProFound (v. 4.10.5)
True positive	132	120	88
False positive	4	1	0

The experiment is described in the text.

It is not difficult to find a peak picking tool; some tools are freely available on the Internet and all MS vendors supply software with their machines that can be used to pick peaks (both manually and automatically). However, it is difficult to find a flexible peak picking tool that is easily incorporated into a PMF pipeline on any platform and which can, e.g. be called from within scripts so that several signal-to-noise levels can be scanned automatically.

It is also fairly easy to find PMF database matching software; several tools are available on the Internet, but none of them integrated with a peak picking tool. It has been our experience that the scoring algorithm presented here is on par with the well-known Mascot (Perkins *et al*., 1999) and ProFound (Zhang and Chait, 2000) tools. This is illustrated in Table 2 where the performances of these three tools on a set of 266 spectra with yeast proteins are shown. The peak lists were produced using the peak extraction tool described in this paper, and filtered using an expert filter with known trypsin autolysis peaks. The database searches were done using a mass tolerance of 200 ppm, allowing one missed cleavage, a fixed cysteine CAM (C_2H_3ON) modification and a variable methionine oxidation modification. Significant hits for the different softwares were defined as P-value <0.05 (Piums), score >66 (Mascot) or Z-score >1.65 (ProFound). These are comparable significance levels. The Piums and Mascot tools searched against SwissProt v. 43.1 (220 438 entries) whereas ProFound searched against NCBInr1004/07/01 (1 313 300 entries) because SwissProt was not available for ProFound. It is harder to do a clear identification in a larger database why the ProFound results are a bit worse than what they had been with SwissProt.

The execution time for analyzing (peak picking, peak filtering and database matching) a batch of 91 samples was 3 min for Piums (our tool) and 15 min with Mascot, when Mascot had the protein database in memory and Piums had a pre-cleaved database. Our tool was thus about five times faster in batch mode than Mascot. The comparison using a pre-cleaved database is relevant since our scoring software is designed for batch processing with many spectra from the same gel: the database is cleaved initially and then used for all spectra so that time is saved from the second spectrum onwards.

All the PMF modules presented here are designed such that they can be used on any platform (for which there exists a C++ compiler) and can be called from within scripts, e.g. in a PMF pipeline setting, or from a GUI, e.g. in an interactive setting. Furthermore, each step outputs lots of information in a structured way, information that can be used in later steps of the process. We believe that this is their strength, because the best PMF performance is achieved when each step in the PMF analysis knows what the other steps have done/will do.

4 CONCLUSIONS

Getting the most protein information out of a mass spectrum requires a system-wide look at the problem. The way peaks are extracted and post-processed has a considerable impact on the success rate in the protein matching. A set of tools have been presented that were designed to simplify the combination of different tools significantly and be flexible enough to fit many different settings (academic, commercial, large-scale and small-scale). The modular design and the use of XML format makes them transparent and easy to use in whole or in combination with other tools, e.g. other peak extraction algorithms or peptide fingerprinting algorithms. The performance of these tools is state-of-the-art, in the sense that we have not seen any other tools that consistently perform the job better, but there are tools that do the job equally well. The presented tools are, however, more modular, more easily scriptable, and more platform independent than any other PMF tools known to us.

REFERENCES

Altschul,S., Madden,T., Schaffer,A., Zhang,J., Zhang,W., Miller,W. and Lipman,D. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Berndt,P., Hobohm,U. and Langen,H. (1999) Reliable automatic protein identification from matrix-assisted laser desorption/ionization mass spectrometric peptide fingerprints. *Electrophoresis*, **20**, 3521–3526.

Breen,E.J., Hopwood,F.G., Williams,K.L. and Wilkins,M.R. (2000) Automatic poisson peak harvesting for high throughput protein identification. *Electrophoresis*, **21**, 2243–2251.

Chamrad,D.C., Koerting,G., Gobom,J., Thiele,H., Klose,J., Meyer,H.E. and Blueggel,M. (2003) Interpretation of mass spectrometry data for high-throughput proteomics. *Anal. Bioanal. Chem.*, **376**, 1014–1022.

Clauser,K.A., Baker,P. and Burlingame,A.L. (1999) Role of accurate mass measurement (±10 ppm) in protein identification strategies employing MS or MS/MS and database searching. *Anal. Chem.*, **71**, 2871–2882.

Eriksson,J., Chait,B.T. and Fenyö,D. (2000) A statistical basis for testing the significance of mass spectrometric protein identification results. *Anal. Chem.*, **72**, 999–1005.

Fenyö,D. and Beavis,R.C. (2003) A method for assessing the statistical significance of mass spectrometry-based protein identifications using general scoring schemes. *Anal. Chem.*, **75**, 768–774.

Fenyö,D., Qin,J. and Chait,B. (1998) Protein identification using mass spectrometric information. *Electrophoresis*, **19**, 998–1005.

Field,H.I., Fenyö,D. and Beavis,R.C. (2002) RADARS, a bioinformatics solution that automates proteome mass spectral analysis, optimises protein identification, and archives data in a relational database. *Proteomics*, **2**, 36–47.

Gay,S., Binz,P.-A., Hochstrasser,D.F. and Appel,R.D. (1999) Modelling peptide mass fingerprinting data using the atomic composition of peptides. *Electrophoresis*, **20**, 3527–3534.

Gobom,J., Mueller,M., Egelhofer,V., Theiss,D., Lehrach,H. and Nordhoff,E. (2002) A calibration method that simplifies and improves accurate determination of peptide molecular masses by MALDI-TOF MS. *Anal. Chem.*, **74**, 3915–3923.

Gras,R., Müller,M., Gasteiger,E., Gay,S., Binz,P.-A., Bienvenut,W., Hoogland,C., Sanchez,J.-C., Bairoch,A., Hochstrasser,D.F. and Appel,R.D. (1999) Improving protein identification from peptide mass fingerprinting through a parameterized multi-level scoring algorithm and an optimized peak detection. *Electrophoresis*, **20**, 3535–3550.

Hjernø,K., Roepstorff,P. and Højrup,P. (2002) Peakerazor: pre-processing of mass lists prior to peptide mass searching. Poster presented at the 14th *Meeting Methods of Protein Structure Analysis (MPSA)*, Valencia, September 8–12.

Karty,J.A., Ireland,M.M.E., Brun,Y.V. and Reilly,J.P. (2002) Artifacts and unassigned masses encountered in peptide mass mapping. *J. Chromatogr. B*, **782**, 363–383.

Levander,F., Rögnvaldsson,T., Samuelsson,J. and James,P. (2004) Automated methods for improved protein identification by peptide mass fingerprinting. *Proteomics*, **4**, DOI10.1002/pmic.20030804.

Magnin,J., Masselot,A., Menzel,C. and Colinge,J. (2004) OLAV-PMF: a novel scoring scheme for high-throughput peptide mass fingerprinting. *J. Proteome Res.*, **3**, 55–60.

Mann,M. (1995) Useful tables of possible and probable peptide masses. In *43rd ASMS Conference on Mass Spectrometry and Allied Topics*, Atlanta, GA, 21–25 May, p. 639.

Pappin,D.J.C., Hojrup,P. and Bleasby,A.J. (1993) Rapid identification of proteins by peptide-mass fingerprinting. *Curr. Biol.*, **3**, 327–332.

Parker,K.C. (2002) Scoring methods in MALDI peptide mass fingerprinting: chemscore, and the chemapplex program. *J. Am. Soc. Mass Spectrom.*, **13**, 22–39.

Parker,K.C., Garrels,J.I., Hines,W., Butler,E.M., McKee,A.H.Z., Patterson,D. and Martin,S. (1998) Identification of yeast proteins from two-dimensional gels: working out spot cross-contamination. *Electrophoresis*, **19**, 1920–1932.

Perkins,D.N., Pappin,D.J.C., Creasy,D.M. and Cottrell,J.S. (1999) Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, **20**, 3551–3567.

Rögnvaldsson,T., Häkkinen,J., Lindberg,C., Marko-Varga,G., Potthast,F. and Samuelsson,J. (2004) Improving automatic peptide mass fingerprint protein identification by combining many peak sets. *J. Chromatogr. B*, **807**, 209–215.

Schittkowski,K. (2003) QL: a Fortran code for convex quadratic programming. *Report*, Department of Mathematics, University of Bayreuth.

Wool,A. and Smilansky,Z. (2002) Precalibration of matrix-assisted laser desorption/ionization-time of flight spectra for peptide mass fingerprinting. *Proteomics*, **2**, 1365–1373.

Zhang,W. and Chait,B.T. (2000) Profound: an expert system for protein identification using mass spectrometric peptide mapping information. *Anal. Chem.*, **72**, 2482–2489.