Statistical and Computational Methods for Comparative Proteomic Profiling Using Liquid Chromatography-Tandem Mass Spectrometry

Jennifer Listgarten‡ and Andrew Emili§¶‖

The combined method of LC-MS/MS is increasingly being used to explore differences in the proteomic composition of complex biological systems. The reliability and utility of such comparative protein expression profiling studies is critically dependent on an accurate and rigorous assessment of quantitative changes in the relative abundance of the myriad of proteins typically present in a biological sample such as blood or tissue. In this review, we provide an overview of key statistical and computational issues relevant to bottom-up shotgun global proteomic analysis, with an emphasis on methods that can be applied to improve the dependability of biological inferences drawn from large proteomic datasets. Focusing on a start-to-finish approach, we address the following topics: 1) low-level data processing steps, such as formation of a data matrix, filtering, and baseline subtraction to minimize noise, 2) mid-level processing steps, such as data normalization, alignment in time, peak detection, peak quantification, peak matching, and error models, to facilitate profile comparisons; and, 3) high-level processing steps such as sample classification and biomarker discovery, and related topics such as significance testing, multiple testing, and choice of feature space. We report on approaches that have recently been developed for these steps, discussing their merits and limitations, and propose areas deserving of further research. ***Molecular & Cellular Proteomics 4:419–434, 2005.***

With the sequencing of the human genome largely complete and publicly available, emphasis in molecular biology is shifting away from DNA sequencing and related problems toward a systematic evaluation of how the myriad of encoded gene products operate together to mediate the biological mechanisms that sustain life, and how these processes become perturbed in response to disease. Comprehensive systems-wide biological studies have been greatly facilitated by the advent of large-scale genomic, proteomic, and informatic technologies, such as DNA microarrays, ultra-sensitive high-throughput protein MS, and robust statistical and machine-learning methods developed for very large datasets. Evaluation, interpretation, and integration of data produced by these respective platforms represent major ongoing challenges and areas of active research.

The field of expression proteomics seeks to answer the following questions: 1) which proteins and variant isoforms are expressed during the lifecycle of an organism; 2) which post-translational modifications occur in each of these proteins; 3) how do these patterns differ in different cell types and tissues and under different developmental, physiological, and disease conditions; and 4) how can biologists make use of this information to better understand the molecular basis for fundamental biological processes as well as for monitoring the course of disease so as to improve clinical diagnosis and treatment (1–3). These questions are made all the more difficult by the complexity of most biological systems, which increases exponentially as one goes downstream from DNA sequence to mRNA intermediates to the protein end-products. While it appears there are likely far fewer genes coded for by the human genome than first anticipated, it is estimated that >60% of the ~25,000 putative ORFs encode more than one splice variant (often tens to hundreds), and these in turn are frequently subject to post-translational modification (4, 5). Moreover, because proteins typically function together as components of dynamic multisubunit macromolecular complexes, the final complexity of a biological system is enormous; hence the search for interesting and relevant proteomic patterns remains a challenging task.

Capillary-scale HPLC-MS/MS (LC-MS) is rapidly emerging as a method of choice for large-scale proteomic analysis (1, 2, 6–12). State-of-the-art LC-MS systems can be used to identify and track the relative abundance of thousands of molecules (6, 7). For standard bottom-up profiling experiments, the molecules in question are peptides derived by proteolysis of intact proteins. For very complex protein samples, such as blood, the peptide mixtures are physically resolved by chromatographic separation prior to injection into the mass spectrometer so as to generate a more-richly informative map, consisting of both the unique elution characteristics (column retention times) as well as *m/z* (mass-over-charge) ratios of individual peptides. Discrete peptides of interest are subject to collision-induced fragmentation followed by database matching for the purpose of sequence identification, while the

From the ‡Department of Computer Science, University of Toronto, Toronto, Ontario M5S 3G4, Canada; §Program in Proteomics and Bioinformatics and ¶Banting and Best Department of Medical Research, University of Toronto, Toronto, Ontario M5G 1L6, Canada

Received, January 20, 2005

Published, MCP Papers in Press, March 1, 2005, DOI 10.1074/mcp.R500005-MCP200

recorded pattern of precursor ion peak intensities can be used to infer the relative quantities of the various proteins between samples. Nevertheless, comparisons of large-scale multivariate proteomic datasets are subject to a number of challenging analytical issues, such as experimental noise, systematic variations between experimental runs, the extreme overall range and dynamic nature of protein levels, and the huge number of measured features (*e.g.* protein levels) of which many are uncorrelated (or spuriously correlated) to the variables of interest.

In this review, we discuss important computational and statistical concepts that should be considered when performing comparative proteomic analyses and outline procedures for processing MS data to achieve more reliable quantitative analyses. As the broader domains of statistical and machine-learning methods are beyond the scope of this review, we limit our examination to studies applying these methods to shotgun profiling datasets. For more general information about statistical learning models, a good reference is Ref. 13. For an in-depth discussion of the technical nature of LC-MS, and the relative merits of different MS platforms, the reader is directed to other reviews present in this issue.

LC-MS-BASED PROTEOMIC PROFILING

LC-MS systems consists of: 1) a chromatography column, which separates peptide mixtures based on one or more physicochemical properties prior to MS; 2) an ionization source, which converts eluting peptides into gas phase ions; 3) one or more mass analyzers, which separate ions on the basis of *m/z* ratios; and 4) a detector, which registers the relative abundance of ions at discrete *m/z*. In MS/MS, precursor ions are recorded in full-scan mode (all *m/z* values), followed by selective ion isolation and fragmentation for sequence identification. MS/MS instruments are operated in an automated alternating scan mode. Two main ionization technologies used are ESI and MALDI (6). Because ESI generates ions directly from solution, it is readily coupled to LC or capillary electrophoresis. In a standard reverse-phase HPLC setup, the column media differentially retards the migration of peptides based on selective hydrophobic interaction affinities. Peptides are then eluted with a gradient of organic solvent and ionized just prior to introduction into the mass spectrometer. LC is well-suited to examining complex biological samples because: 1) peptides with the same nominal *m/z* are less likely to be introduced at the same time, reducing ambiguity; and 2) with fewer competing ion species, fewer artifacts arise due to ion suppression or ion-ion interference.

A non-LC-based MS platform commonly used in profiling studies is SELDI (a refinement of MALDI), in which subsets of proteins in samples are selectively pre-adsorbed onto various proteomic chips (consisting of differing binding surfaces, such as hydrophobic or metal-binding surfaces) (8, 9).

The quality of profiling studies is determined by the overall sensitivity, detection coverage, dynamic range, fragmentation efficiency, mass resolution, and accuracy (6). Femptomole or better detection limits are commonly attained with LC-MS, even with mixtures exceeding several thousand components (7). With SELDI, much of the sample is discarded (not selectively bound to the matrix), whereas LC-MS enables full sample throughput, while providing the added opportunity for protein identification and mapping of post-translation modifications.

PARADIGMS FOR INFORMATION EXTRACTION FROM LC-MS DATASETS

High-throughput MS/MS offers a powerful means of analyzing biological samples. However, MS/MS is time consuming, adding an order of magnitude in time to a profiling experiment over full-scan precursor MS profiling. Much time is wasted repeatedly resequencing the same peptides (even with data-directed experiments) across different samples. Additionally, due to the biased under-sampling nature of MS/MS ion selection (15), it is impossible to sequence every peptide in a sample in a single-pass analysis. In the context of a comparative LC-MS profiling analysis of multiple samples, such as serum profiling for disease biomarker discovery, one can imagine avoiding some of the less-desirable properties associated with MS/MS. For example, a reference database of previously sequenced peptides ion species could be built up. Given a new sample, one could imagine running LC-MS alone (*i.e.* without performing CID), obtaining a global signature pattern of peptide ion abundance, and then scanning the database for discrete pairs of time*, m/z* values to predict peptide identity. In this way, the same peptides need not be resequenced, experimental run-time is reduced, and differential expression can be more readily uncovered (1).

However, such an approach is complicated by a number of factors: 1) the LC time axis needs to be corrected to account for spurious deviations in peptide elution times across different experiments; 2) there may be confounding overlap of peptides across the (time*, m/z*) space; 3) LC-MS systems are subject to considerable noise and variability that is not fully characterized or accounted for; and 4) differences in overall sample composition, leading to differential ion context, may affect the apparent signal intensities recorded for peptides common to multiple samples.

To address these issues, the following tasks need to be resolved: 1) corrective alignment along the experimental LC time axis so that times are comparable across experiments; 2) combining replicate experimental LC-MS datasets so as to improve the overall signal-to-noise ratio; 3) developing statistically sound methods for distinguishing signal from background; 4) systematically studying sources of variability and signal characteristics inherent to LC-MS data and modeling them so as to take them into account (for example, systematic variation in peak amplitude and width, signal linearity, and background artifacts should be investigated); 5) developing algorithms to detect and quantify peptide ion peaks in LC-MS data; and 6) developing probabilistic algorithms to uncover

TABLE I
Sequence of operations for processing LC-MS datasets

1	Quantization of *m/z* values
2	Signal filtering and background subtraction
3	Amplitude normalization
4	Peak detection and quantification
5	Data transformations/error models
6	Alignment in time (and *m/z*)
7	Classification algorithms and biomarker discovery

interesting peptide patterns and to match new patterns to previously discovered ones.

In the aforementioned approach, peptide peaks are extracted and then identified, and only after these steps are performed does one pose queries related to the particular experiment at hand. An alternative approach to exploiting LC-MS data is to take a signal processing perspective. For instance, rather than starting with peak detection and peptide identification to find disease biomarkers using LC-MS patterns, the data could be treated as a signal matrix, allowing the application of established methods in signal processing, statistics, and machine learning (13, 16, 17) to tease out interesting and relevant patterns from the data. Nevertheless, many of the previous tasks, such as time alignment, background detection and correction, and accounting for systematic variations in signal amplitude, are relevant to this latter approach.

Several groups (1, 10, 18) have developed working systems to address the main problems described above. The system of Radulovic *et al.* (18) performs filtering, normalization, peak detection, quantification and alignment, and then classification; they also establish linearity in LC-MS peak signal with peptide concentration. Although no algorithm details are provided, Kearney and Thibault (1) report a system for peptide detection and data alignment, which identifies the *m/z,* retention time, charge state, and maximum intensity of the principle peptide isotope. Wang *et al.* (10) performed baseline subtraction, peak detection, isotope detection, alignment, and quantification and demonstrated a linear LC-MS peak signal response to increasing peptide concentration in a complex mixture over a fairly large dynamic range.

Several studies of LC-MS have focused on particular low/mid-level data processing steps such as noise reduction, error models (which model the variance of the peptide abundance), or alignment in time, while other studies have been devoted strictly to the use of MS signal for sample classification. For studies attempting to cover the entire set of tasks, a typical approach for information extraction from LC-MS datasets typically involves a sequence of operations as listed in Table I, each treated in a largely independent manner from the others. The optimal order in which these should be performed so as to obtain the most information possible remains unclear. For instance, one could align the signals before or after peak detection, one could normalize before or after alignment, *etc.* There are several sequences of operations that make intuitive sense, but that differ significantly from one another. The optimal ordering depends in part on our ability to perform each task. For example, if alignment can be performed optimally on the raw data, without normalization or peak detection, then it may be desirable to do this first, potentially allowing for better normalization and peak detection. Alternatively, if one can not align the raw signals well, then it may be desirable to normalize and find peaks before alignment. Lastly, it may be desirable to approach some of these subtasks together, for example as in the continuous profile model (CPM),[1] which aligns and normalizes abundance simultaneously, removing the necessity to choose one before the other (19).

LC-MS DATA PROCESSING

Operationally, low-level processing involves raw LC-MS signal, while mid-level processing occurs after some basic preprocessing, such as *m/z* quantization, filtering and background subtraction (*i.e.* low-level processing). In contrast, high-level processing is applied to data that has been fully massaged for use in conjunction with machine-learning techniques or more traditional statistical techniques such as significance testing of individual features (*e.g.* peptide abundance). The substantial collection of methods developed for processing nonchromatographic MS data (*e.g.* MALDI and SELDI studies) is in many cases transferable to LC-MS data (which can be viewed as a time series of static MS spectra). Most of the low- and mid-level processing methods reported to date, however, have been performed parenthetically as a means to the larger goal of sample classification or biomarker discovery and hence have not been rigorously studied.

LOW-LEVEL PROCESSING—QUANTIZATION OF *M/Z* VALUES

In its most raw form, the full-scan MS spectra obtained from an LC-MS experiment generally consists of a table of values consisting of: i) scan (spectra) number; ii) LC retention time; iii) precursor ion *m/z* value; and iv) ion abundance. There is a one-to-one mapping such that the scan number is simply an enumeration of the ordered time points. For more easy data manipulation, LC-MS datasets can be converted into a matrix format, with columns representing *m/z* values and rows representing time. Entries in the matrix represent the relative abundance values at each combination of time and *m/z* bin. This matrix formation involves binning nominal *m/z* values, because retaining all possible values would lead to a huge,

[1] The abbreviations used are: CPM, continuous profile model; GC, gas-phase chromatography; PCA, principal component analysis; TIC, total ion current; DP, dynamic programming; COW, correlated optimized warping; BPC, base peak chromatogram; PARAFAC, parallel factor analysis; HMM, hidden Markov model; DTW, dynamic time warping; CV, coefficient of variation; FPR, false-positive rate; FDR, false-discovery rate; ROC, receiver-operator characteristic; RF, random forest; LDA, linear discriminant analysis; SVM, support vector machine; QDA, quadratic discriminant analysis; PPC, peak probability contrasts; NSC, nearest shrunken centroid; PLS, partial least squares.

sparsely populated matrix, while time values can normally be left unchanged because these are usually not too numerous and because many *m/z* values typically correspond to a given time point.

An issue to consider here is how to bin the *m/z* values. For example, one could opt for evenly spaced bins in either native or log *m/z* space. An optimal bin width would be large enough to keep the matrix tractable and not too sparse, but small enough that individual *m/z* values remain informative (*i.e.* not collapsing the information too much); this trade-off depends on the MS instrumentation used. For instance, Radulovic *et al.* (18) decided to round *m/z* values to the nearest integer, consistent with the nominal mass accuracy of the ion trap instruments used in their study, while Wang *et al.* and Anderle *et al.* (10, 20) used evenly spaced bins of width 0.2*Th* to exploit the increased accuracy of their instruments. In a study of gas-phase chromatography (GC)-MS data by Stein (21), bin widths were chosen as a function of the measured mass accuracy and resolution and increased linearly with *m/z*. In Ref. 22, LC-MS peak-widths (ignoring time) were reported to be reasonably constant in log *m/z* space by Tibshirani and coworkers. No methods have been reported for evaluating optimal bin width, nor for determining the sensitivity of further calculations to this parameter.

SIGNAL FILTERING AND BACKGROUND SUBTRACTION

Because LC-MS is subject to background chemical and electronic noise, together with systemic contaminants in the LC mobile phase (column solvent) (23, 24), methods for noise reduction and signal enhancement are commonly used. Signal filtering is a mature field from which a variety of techniques are applicable. The theory of digital signal processing is based on the assumption that data were sampled at regular time intervals, which is not necessarily the case for many LC-MS experiments. Filtering may nevertheless be useful provided extra care is taken to account for this. Conceptually, signal filtering and baseline subtraction can each be performed in both the time (scan header) and *m/z* domains. Two approaches applied to date include: i) subtracting a fitted, additive baseline model; and ii) using digital filters to smooth and enhance the MS signal. In certain cases, filtering is performed first, and then a modeled baseline is subtracted (10).

Various filters for data smoothing along the LC time axis have been implemented (25). These include simple “moving average,” median, and moving geometric mean filters, and the Savitzky-Golay filter, which preserves high-frequency content by fitting a higher-order polynomial to the data over a local window (26). For example, Wang *et al.* observed well-defined peaks and baseline after applying the Savitsky-Golay filter to LC-MS data (10). Data points belonging to the baseline were then hand-picked, fit with a low-order polynomial, and subtracted from the original data, together with a second application of the Savitsky-Golay filter for added peak smoothing. Nevertheless, manual delineation of background is a subjective, tedious, and error-prone process, and inconsistent with high-throughput analysis.

Bylund *et al.* have noted that taking the second derivative of raw MS signal, followed by matched filtering*,* whereby signal (in this case the second derivative of the signal) is cross-correlated with a Gaussian template, reduces background noise and enhances peaks. This smoothing is similar to application of a low-pass “top hat” filter, as is done on SELDI-MS data along the *m/z* axis in (27) in which signal frequencies above some threshold are thereby completely removed from the data. Note that by taking the second derivative of the signal, nonlinear noise is actually amplified (and hence the need to filter afterward), while with matched filtering*,* noise (specifically white noise) adjusts to the template frequency, making it harder to identify. On the other hand, Radulovic *et al.* (18) reported a two-step procedure for noise reduction/binarization of LC-MS signal. First, a “moving average” filter (five-scan header width) is used to smooth the dataset across discrete *m/z* bins. Then peak intensities exceeding a pre-defined threshold, *T* (related to the trimmed mean or median intensity of one *m/z* bin across all scan headers), for *N* consecutive scans are selected as being signal, with the rest of the intensities deemed to be noise. Processing of negative control spectra acquired during LC-MS analysis of solvent alone revealed few false-positive peaks. However, the number of missed genuine peptide peaks (false-negatives) is quite sensitive to the selected processing parameters (values of *T*, *N*, or *M*), and peak detection efficiency has not been fully optimized.[2]

Wagner *et al.* (10) have made use of an iterative, nonparametric, local regression smoother (a robust *loess* smoother) to model the baseline in MALDI-MS datasets. Because distinct regions of *m/z* were different in nature, empirical selection of the size of the smoothing window for each region was necessary and ranged from 1% of the total number of *m/z* values for regions with small *m/z* to 70% for regions with larger *m/z* values. In contrast, Baggerly *et al.* found a single sinusoidal baseline noise component in their MALDI dataset, which they speculate was produced by use of an alternating current power source (28). The noise frequency was estimated by Fourier transform of a hand-selected data region, and then eliminated by subtracting out a sinusoid of this frequency. Following this procedure, the residual baseline was modeled with a modified local minimum at each *m/z* value and them removing the modeled baseline.

Bylund *et al.* developed a unique “orthogonal background subtraction” approach for LC-MS baseline correction (25), wherein principal component analysis (PCA) is applied to time vectors (one vector per time point, over all *m/z*) from a region in the dataset expected to consist solely of background. The noise subspace was then characterized by taking the top principal components, and noise is removed by subtracting

[2] A. Emili, unpublished observations.

the components of the data that lie in this subspace. While such a model may prove useful, it is not clear whether it is appropriate for LC-MS analysis; PCA operates in a global way (across all time points in this case) and therefore does not allow for noise characteristics to change over time.

Concerns have been raised about the suitability of data filtering followed by parametric fitting (*e.g.* based on a peak model) (29), and thus one must carefully consider the ultimate goal of a LC-MS-profiling experiment when performing tasks in sequence. To our knowledge, no systematic comparison of the effects of various filtering/background subtraction techniques on MS data integrity has yet been reported, and it is advisable to investigate the consequences on downstream analysis (*e.g.* classification, peak detection). Because the filtering efforts published to date were performed on only one data dimension (*m/z*), it will be interesting to see if filtering independently in both axes (time and *m/z*) or simultaneously is more beneficial.

MID-LEVEL PROCESSING—PEAK DETECTION AND QUANTIFICATION

In certain situations, it may be desirable to detect and quantify two-dimensional peptide peaks in LC-MS signal for use as input to classification algorithms, biomarker discovery, or global proteomic comparisons using a unified reporting schema. Extracting peaks from LC-MS signal both reduces the dimensionality of the data, which can simplify downstream analysis, and assigns intuitive meaning to data features. (Possible drawbacks of this approach are discussed further below.)

Radulovic *et al.* used an iterative coarse-to-fine strategy to extract two-dimensional (in time and *m/z*) peaks from LC-MS data (18). Neighboring points in the data matrix deemed to be signal (rather than noise—see previous section on filtering) were combined to form peaks at the coarsest level, and then iteratively through each of the more refined levels, with a bisection method used to avoid spurious peak mergers. Peaks were quantified by summing individual grouped feature intensities. On the other hand, Wang *et al.* detected LC-MS peaks based on coinciding local signal maxima, in time and *m/z*; local maxima are defined as an increase in ion abundance greater than a prespecified threshold over a predefined range (10). Peaks were then quantified either by summing intensity over the component elution time or based on the maximum peak height. Unfortunately, the authors do not describe how the component elution time was determined. Similar techniques were used in Refs. 20 and 30.

Yasui *et al.* (31) defined peaks (in SELDI-MS data) as *m/z* elements exhibiting higher intensity than the *N* nearest neighbors, with *N* chosen empirically, with an added constraint that peaks have a higher intensity than the "broader" neighborhood as calculated by a local linear super-smoother method. In an effort to reduce dataset misalignment problems, *m/z* values within ±0.2% of these peaks were further selected as additional peaks. Although peak finding was not the emphasis of the study of Tibshirani *et al.* (22), these authors used a similar routine to scan for *m/z* peaks in SELDI/MALDI datasets exhibiting a higher intensity than a prespecified number of closest neighbors. In contrast, Randolf *et al.* made use of multiscale wavelet decomposition to detect peaks in MALDI-MS data (32), trying to avoid *ad hoc* decisions pertaining to thresholds and filter parameters. The wavelet decomposition provides a breakdown of the signal according to scale (and location), and by taking the derivative at each scale these authors detected scale-specific peaks. In a histogram of peak locations extracted in this way from many samples, locations with high counts were considered as evidence of true peaks, although the authors did not offer a method for choosing an optimal scale.

Instead of attempting to find peaks, Idborg *et al.* applied a curve resolution approach developed by the Chemometrics community for processing GC-MS profiles (see Ref. 33 for an overview of Chemometrics analysis techniques) to extract the major spectral components in LC-MS profiling of urine (34, 35), a notably simpler mixture than tissue or serum. In this manner, a data matrix, X, defining a single LC-MS experiment, with rows corresponding to time points and columns to *m/z* values, was resolved into a set of "pure" spectral profiles. These profiles were stored as column vectors of length m in the matrix, S, with the complete set of profiles approximately spanning the entire space of all *m/z* spectra generated across the various time points. The relative amount of each spectral profile present at a given time point was then estimated in the "pure" concentration profiles, stored as t row vectors of length a in the matrix C. Unexplained variation was represented by matrix E, and $X = CS^T + E$. An iterative method was used to solve for C and S, starting from an initial set of "key" spectra and using constraints related to the non-negativity of the concentrations (*i.e.* related to the fact that one can only add in physical components, not remove them), to update the spectral profiles. However, Idborg *et al.* do not describe how to select the number of spectral profiles, although various approaches to this problem are provided in Ref. 33, as well as other algorithms used for spectral resolution. Because use of these techniques has been largely restricted to relatively simple mixtures, it is not clear how well these methods will scale to more complex proteomic samples.

Overall, peak detection has generally been performed in a rather *ad hoc* manner, with little evaluation of the effectiveness of the various methods or parameter choices. The algorithms employed to date make no use of *a priori* or learned information with regards to peak shape, along either the time or *m/z* dimensions, and in some cases ion intensity values are only exploited very indirectly. Rather than retaining abundance information, peaks are frequently binarized (18, 31). Radulovic *et al.* do not motivate this decision, while Yasui *et al.* state that it helps to overcome noise in the signal. Such a step is lossy and is likely suboptimal for downstream analysis. Incorporating richer information would likely improve analytical performance, albeit at the cost of more computation. The underlying methodologies of machine learning and statistical techniques are intended to account for random variation caused by noise, and their performance is likely deteriorated by using them with binarized MS data. A less-extreme approach, and one retaining more information, would be to apply filtering and/or baseline subtraction as well as intensity normalization rather than binarization. It has been suggested that at present LC-MS is still not generally a quantitative science (36). Peak detection and quantification, even if done optimally, does not guarantee linearity of peak signal relative to analyte concentration due to possible ion suppression effects, although compelling evidence of linearity of extracted LC-MS peak intensities, at least for spiked reference proteins, has been established using certain data-processing methods and technological platforms (10, 18).

DE-ISOTOPING, CHARGE DECONVOLUTION, AND PEAK MATCHING

Additional considerations relevant to LC-MS proteomic analysis are the charge spread on the ions, and also stable isotope variants. ESI results in variable net positive (or negative) charge. Multiply charged molecules are observed, with signal spread out along the *m/z* axis in a charge state envelope. Under typical conditions, mixtures of singly, doubly, and even triply protonated precursor ions are commonly observed with tryptic peptides. If the sample analyzed is simple, it is possible to deduce related *m/z* peaks. Charge deconvolution relies on determination of isotope shoulders, resulting from the presence of heavy isotope variants of carbon and other atoms. Roughly ~1% of naturally occurring carbon exists with seven (as opposed to the more common six) neutrons (^{13}C *versus* ^{12}C); this signature dominates other isotopes because ~50% of the mass of a typical peptide is carbon (37). Although isotope variants are chemically identical, the heavier isotope sister ion peaks exhibit greater apparent *m/z* than the predominant monoisotopic peak (all ^{12}C). Based on the mass differential, one can deduce the charge state of multiply charged ions.

Wang *et al.* and Anderle *et al.* applied a de-isotoping step (to assign isotope patterns) before peak quantification (10, 20). Though few details are provided, the algorithms were apparently based on cross-correlating the observed peak envelopes to reference isotopic tables, with the highest-scoring match identifying the most probable isotope shoulders. One can imagine that probabilistic modeling techniques, such as hidden Markov models (HMMs), may significantly improve upon this template-matching scheme. In contrast, Tibshirani *et al.* opted to smooth MALDI/SELDI data along the *m/z* axis as a simpler alternative to deconvolution (22). Extremely complex samples may prove to be less amenable to full de-isotoping.

Peak matching is another related topic relevant to quantitative proteomic comparisons. To measure reproducibility of peptide signal, experimental peaks must be matched across LC-MS datasets. Naive methods, based on simple proximity (in time or *m/z* space), are reported to be effective (10, 20, 39). For instance, Radulovic *et al.* used MS/MS-derived sequence identities to verify the correct matching of ~200 putative peptides across multiple samples (18). However, given that MS/MS targets prominent peaks, this assessment is likely biased. Anderle *et al.* (20), on the other hand, found it necessary to remove ~2% of data points—presumed outliers thought to be attributable to mismatching (20). Incorporation of prior knowledge of peak shape, instrument *m/z* drift, and a more-probabilistic formulation might significantly improve the effectiveness of peak detection, quantification, and matching.

DATASET ALIGNMENT AND COMPARISON

While the accuracy of *m/z* measurements obtained with properly calibrated instruments is typically very good, LC fractionation is inherently variable (*i.e.* considerable dispersion in peptide retention times). Elution patterns can become distorted (locally compressed and/or expanded in complex, nonlinear ways) by differences in chromatography performance due to changes in ambient pressure and temperature. Even under ideal conditions, MS duty cycles are finite and sampling is not necessarily constant, resulting in spectral capture at different time points across an eluting peak even between repeat analyses. This variation can affect peak discrimination and global proteomic comparisons. Thus, to maximize the benefits of LC-MS, one needs to deal with the inherent variability in the time axis (*i.e.* recorded retention time or scan headers). In certain cases drift may occur along the *m/z* axis as well, although this is far less of a problem than variations in time. Time and *m/z* axes can be aligned independently or simultaneously, though the latter has not been reported in the literature and would be more easily applied after peak detection. Furthermore, if aligning in time only, one may wish to use scalar time series rather than the vector time series most readily available from the data (*e.g.* total ion current (TIC) as scalar time series *versus* a vector of all *m/z* values at each time point), or even more general representation schemes, such as a reduced-dimensionality vector time series as obtainable for example by PCA.

Alignment algorithms typically involve either: i) maximizing some objective function over a parametric set of transformations (usually linear); or ii) nonparametric alignment, by way of dynamic programming (DP); or iii) some combination of these methods (*e.g.* piecewise linear transformations). Some of the main differences among algorithms are: i) whether alignment is performed before or after peak detection (*i.e.* using either continuous signal or peaks); ii) whether or not signal amplitude is used; and iii) whether or not changes in scale are corrected for (*i.e.* allowing for interplay between these two types of corrections). Most algorithms used to date require a template, specified *a priori,* to which all time series are pre-aligned. Because suboptimal template choice could result in poor alignments, it may be desirable to avoid this.

Nielsen *et al.'s* correlated optimized warping (COW) algorithm (40) and modifications of it were used by Bylund *et al.*

(41) to align chromatographic data by dividing the time axis into segments, and then performing a linear warp within each segment to optimize overlap while constraining segment boundaries to maintain agreement. An objective function defined the optimal set of transformations based on the sum of correlation coefficients or covariance between data segments in pairs of samples, and it was maximized by way of DP. COW can be applied to both scalar and vector time series by defining the correlation or covariance appropriately. Use of more than one data vector component (*e.g.* multiple *m/z* bins rather than the TIC) produced more stable alignments with respect to variation of free parameters such as maximum allowable warp (40). Nielsen *et al.* established visually using artificial chromatograms that COW is robust to varying peak numbers, heights, and widths, and is superior to a global linear warp (40). An interesting evaluation is provided in Ref. 41, where PCA was performed on the base peak chromatograms (BPC). The amount of variance explained by the top two principal components was 70% before alignment and 98% afterward. Similarly, explained variance went from 60 to 97% with a seven-component parallel factor analysis (PARAFAC—a generalization of PCA to three-way data), indicating a reduction in the major sources of sample variation.

Radulovic *et al.* (18) approach to alignment in time used binarized data and one alignment per data block. The data matrix was divided into five equal *m/z* partitions and six equal time partitions, creating 30 blocks. One experimental dataset was warped to match the other by applying a linear transformation with offset to each block and constraining neighboring blocks to have similar transformations. Monte Carlo maximization was used to find the optimal set of linear transformations as defined by a “data overlap” objective function. Finally, a *post hoc* “wobble” of peaks in time was applied to compensate for residual peak drift. Empirical assessment of feature-wise overlap of the datasets implied a considerably improved alignment (18), although the authors did not use an objective measure because their evaluation was based on the function being optimized. Another drawback of the proposed method is that it works only with binarized data, and is therefore sensitive to choice of binarization threshold, and does not exploit signal amplitude, which ultimately may be more informative.

Randolf *et al.* used coarse scale-specific peaks, extracted by multiscale wavelet decomposition, to align MALDI data along the *m/z* axis (32). Dominant peaks (above some threshold) were used to compute a single optimal shift for all peaks, and thus the alignment is not very flexible as it does not allow for even a simple linear stretch or compression. It is also not clear whether or not features detected at different scales should be aligned differently or could leverage one another in alignment. In contrast, Idborg *et al.* do not explicitly align their datasets, but compare detected components derived by curve resolution, using cross-correlation, and shifting individual components to account for constant time shifts between experiments. Components correlating above some threshold are said to be identical (34, 35).

We have recently developed the CPM, an HMM-based model to do multiple alignments of time series (for continuous-valued output, such as the abundances in an MS experiment) using LC-MS TIC traces (19). In the context of the CPM, one can think of the HMM (42, 43) as containing a series of hidden states, each of which represents some underlying “true”’ or canonical time, to which each scan header in the TIC is ultimately assigned. The alignment in time is dictated by which scan header gets (probabilistically) mapped to which hidden state. The states are called “hidden” because until the algorithm is run on the data (*i.e.* until the model is trained), we do not know which scan headers map to which states. In addition to the hidden time states mentioned, hidden states are also augmented by “scale” states, which allow scaling of the TIC amplitudes locally in time. Use of the model after training provides a mapping to both time and scale states, thereby performing alignment and normalization concurrently. Training, whereby the best parameters for the HMM are found, is performed by maximum likelihood (*i.e.* the objective function is defined to be the likelihood of the data under the HMM probabilistic model framework) by way of Expectation-Maximization (44). Both training and later use of the model (*i.e.* deduction of which scan headers map to which hidden states) are performed efficiently in HMMs by use of DP. The CPM has the advantages that no template is required, all experimental TICs are aligned simultaneously (leveraging the information across experiments), normalization local in time is concurrent with alignment, and the model is probabilistically formulated. As pointed out in Ref. 19, the algorithm can be extended to use vector time series rather than TICs and also to allow alignment of nonreplicate data (see end of this section).

The classical algorithm for aligning time series is dynamic time warping (DTW), a DP-based approach that originated in the speech recognition community as a robust distance metric between pairs of time series (45). DTW aligns one time series to a specified reference series. It is closely related to COW, except instead of moving only nodes (time segment boundaries) around, every data point can be moved; thus, transformations are not restricted to piecewise linear. It is likely that Bylund, Nielsen *et al.* avoided use of this less-restrictive model in order to reduce computational costs and avoid overfitting (40, 41). In contrast, Wang *et al.* used DTW on LC-MS data, but constrained the analysis to no more than 200 *m/z* bins so as to make it computationally practicable (10).

Hierarchical clustering was used in a novel way by Tibshirani *et al.* (22) to align MALDI/SELDI spectra along (log) *m/z* space after peak detection. Input to the hierarchical clustering algorithm is a list of putative candidate extracted peaks as well as the Euclidean distance between peaks in log *m/z* space. After clustering was completed, the dendogram was cut off at an empirically determined level, with the mean *m/z* of each cluster defining an individual peak.

A brief description is given for an alignment in *m/z* space for SELDI data by Sauve *et al.* (27) who used a tree, built for example by hierarchical clustering of samples, to guide the progressive warping of related experiments together. Although few details were reported, presumably the algorithm starts by aligning the two closest samples, forming a single pseudo-sample, to which the next closest sample is aligned, *etc*. Such a method removes the need for a prespecified template, but is likely adversely affected by the fact that the distances between spectra are measured before alignment, and hence are largely meaningless. It is also not clear that starting with the two closest spectra is any less arbitrary or more effective than other approaches.

As one moves from a local alignment perspective to a global one, a bias-variance trade-off comes into effect. That is, with more data constraining the transformation, more stable alignments result. The optimal trade off is determined by the informativeness of the LC-MS signal used and the type of misalignments present. In terms of what data are best for use in alignment, there are a number of choices. To date, these have largely been limited to the TIC, BPC, and to individual (or sets of) extracted *m/z* ion chromatograms. In practice, one could select ion chromatograms based on the highest ion count, or highest sum of second derivatives along the *m/z* axis, or rather use some dynamic binning of *m/z* such that the ion signal is evenly distributed to attempt to extract a smaller number of informative *m*/z bins rather than simply choosing evenly spaced ones. Alternatively, one could apply a dimensionality reduction technique, such as PCA, on the *m/z* space, with the aim of using a smaller number of features that are still informative (in this case, the features would be "eigen-*m*/z," that is, pseudo-*m/z* bins made up of linear combinations of the original *m/z* bins).

Whether one uses piecewise linear transformations in small regions, such as in Refs. 18 and 41, or more flexible alignment schemes, such as in Refs. 10 and 19, ultimately may have little importance, so long as the overall transformation is not restricted to a global linear warp. Incorporating local scaling simultaneously with alignment may also prove to be advantageous, as reported in Ref. 19. Another issue to consider is whether to align *m/z* bins individually, together, or somewhere in between (*i.e.* in a smoothly varying way). The issue of whether one should do alignments before or after peak detection has not been clearly answered. Assuming it were possible to correct the LC time axis before peak detection, one could better leverage the information encoded across aligned datasets to achieve more reliable and sensitive peak detection. Historically, with LC-MS data, researchers have concentrated on correcting the time axis, ignoring the *m/z* axis. However, corrections are commonly performed along the *m/z* axis in SELDI/MALDI experiments, suggesting it may be desirable to do so with LC-MS data (though this may be instrument-dependent). Alignment algorithms are typically formulated to work on datasets that are very similar to each other. However, if one knows *a prior* that the samples may differ significantly in a few (unknown) locations (by an unknown amount), for example in comparisons of cancer and noncancer serum, then this should be incorporated into the model, as suggested in Ref. 19. This should improve the overall performance of alignment algorithms and may be a fruitful direction to pursue.

DATA NORMALIZATION

MS signals are frequently corrupted by either systematic or sporadic changes in abundance measurements. That is, overall peak amplitudes measured in one replicate may be elevated with respect to another, and may also have systematic changes within an experiment, across time, due to a change in column or ESI performance. In such cases, the data need to be normalized to make the experiments comparable. The simplest approach would be to multiply all abundance values in one experiment by some constant factor, but in general it may be necessary to perform more detailed corrections. Normalization of MS data can be performed either by coercing *m/z* intensity values to be comparable across experiments (low-level processing), or by altering peak abundance to be comparable (mid-level processing). In general, one aims to normalize not only replicates, but also experimental data of distinct biological origin, such as serum profiles from cancer patients and healthy case controls. The underlying assumption behind normalization is that the overall MS abundance of either all features (peaks or time-*m/z* pairs), or subset(s) of these, should be equal across different experiments. Given this assumption, one can determine the ratio of overall abundance of a chosen set of features between two experiments for use as a multiplicative correction factor, and then normalize an entire set of experiments by arbitrarily choosing one of them as a reference to which all others are normalized.

Global normalization refers to cases where all features are simultaneously used to determine a single normalization factor between two experiments, while local normalization refers to cases where a subset of features are used at a time (different subsets for different parts of the data). Locality can be defined by, say, similarity in *m/z* values, time (scan headers), or abundance (peak intensity) levels. For example, in an abundance-dependent, local normalization, peaks of similar abundance within the same MS experiment would be scaled in a similar way, while peaks of different abundance are scaled in a different way. If the mean of all features is made to agree across all experiments, it is referred to as a global mean normalization, as for example used by Sauve *et al.* (27). By plotting the point-wise log ratio of matched features between datasets *versus* either *m/z* or abundance, Sauve *et al.* (27) visually established that no trend existed along either the *m/z or* intensity axes (27) and hence that the normalization method need not take these factors into account. While several groups have opted for global abundance normalization, in the case of LC-MS data it may be necessary to normalize locally in time (19), because chromatography can produce irregular fluctuations in signal.

Many of the normalization techniques applicable to LC-MS data have also been applied to the results of microarray experiments (46). With gene expression profiles, the genes used for normalization have sometimes been restricted to so-called "housekeeping" genes presumed to remain constant across the experimental conditions. An analogous concept was applied to LC-MS data by Wang *et al.* (10), whereby a constant intensity ratio between pairs of experiments was computed based on reference peaks. These authors noted, however, that the use of all detected peaks provided similar results. Baggerly *et al.* likewise considered using "housekeeping" peaks to normalize, but stated that they were unable to find stable peaks across experiments (28).

Anderle *et al.* (20) opted to normalize multiple LC-MS samples to a single reference dataset, using median global normalization over pairs of matched peaks, while Wagner *et al.* and Baggerly *et al.* (28, 39) chose global mean normalization for processing MALDI data. Radulovic *et al.* also used global normalization, with each dataset multiplied by *K*, such that the total number of intensity values exceeding some predefined threshold was set to equal the somewhat arbitrarily chosen value 100 (18). In contrast, Tibshirani *et al.* (22) normalized their MALDI/SELDI spectra by mapping the 10th and 90th percentiles to 0 and 1, respectively (linearly interpolated between). In our own recent study (19), TIC traces were normalized collectively in conjunction with dataset alignment, leveraging information contained across all experiments simultaneously. Normalization was done locally at each scan header (but globally across *m/z*), with the constraint that neighboring scan headers have similar scaling factors.

Normalization is often evaluated by calculating the coefficient of variation (CV) between peaks across different experiments after normalization. While reasonable CVs (*e.g.* <30%) are commonly reported, a comparison to CVs from prenormalized data is often not provided. Moreover, because no systematic comparison of these various normalization techniques has been reported, it is difficult to assess their relative merits. While Sauve *et al.* reported no abundance-dependent artifacts with SELDI data (27), it will be interesting to see if this holds more generally across data sets, and also for LC-MS data.

DATA TRANSFORMATIONS AND ERROR MODELS

One ultimate goal of many profiling studies is to find proteomic patterns that can discriminate between different biological conditions. In order to properly assign statistical significance to candidate biomarkers, or any changes in apparent protein abundance, it is important to understand the patterns of variability, such as how the variance of measurements correlates with abundance level. When using traditional statistical approaches, such as a *t* test or related measure, one needs to estimate the variance, which can be difficult to do if few replicates are available (as is typical of many LC-MS studies). Error models can be devised that leverage the variance estimates across abundance levels (20) and can be incorporated in the development of machine-learning algorithms for sample classification, possibly leading to improved performance. The log transformation of peak intensities is frequently taken, with some underlying, unarticulated consensus that the log transformation is a good thing (27, 28, 34, 47). Error models can help us to understand why this may (or may not) be the case. Indeed, log transformation converts multiplicative error into additive error and therefore stabilizes variance (20). Nevertheless, Satten *et al.* (48) offered that taking the log of MALDI spectra makes it almost impossible to visually discern peaks and harder to tease apart noise from background, albeit without a statistical basis for this assertion.

To examine patterns of variation and to deduce the variation attributable to sample preparation, Anderle *et al.* conducted a well-controlled LC-MS study, borrowing established parametric models of heteroscedasticity (*i.e.* unequal variance, in this case, across peaks with different abundance levels) from the microarray community. Human serum was fractionated into 40 samples (after removal of the most-abundant proteins and following tryptic digestion), with half of these analyzed directly by LC-MS, and the other half recombined and again resplit before analysis. The variation in the amplitude of matched peak intensities formed the basis of their study. A variance model, ${\sigma^2}_x \approx \alpha\mu^2 + \beta\mu$, was fit to the observed intensity values (with true mean abundance μ and constants α and β) and observed visually to be appropriate. The quadratic term dominates at higher-intensity levels resulting in a constant CV. Fitting the pooled and individual datasets to this model, it was shown that the pooled samples exhibited a CV of 11%, while the individually processed samples had a CV of 20%, suggesting much variation is attributable to sample preparation. Anderle *et al.* also report that application of this error model to two randomly divided subsets of individually processed data, in conjunction with *t* tests on each of the matched peaks between the groups, resulted in far fewer false-positives as compared with no data transformation. The number of false-negatives was not assessed, however, and could be high. Moreover, it would be beneficial to evaluate this model with other LC-MS datasets to ensure that the patterns of variability are generally valid.

Satten *et al.* (48), on the other hand, take the view that "all identification should be made using only features that well exceed a noise threshold, to ensure that the resulting classification algorithm has scientific validity." While these concerns are warranted, the emphasis appears to be in the wrong place. Classification algorithms, when used properly, for example in the context of Bayesian methods, or using cross-validation (with feature selection inside the cross-validation loop (49)), will not assign any importance to random structure in the data. Conversely, the approach offered by Satten *et al.* (48) can cause loss of small, but significant signals, reducing the efficacy of profiling. They also state that "the goal of analysis [is] to show that standardization and de-noising algorithms retain sufficient information to allow categorization." We counter that data processing should improve information availability. Otherwise, faced with more challenging tasks, unnecessarily conservative processing steps may obscure the answers we seek.

Stein *et al.* mention that event-counting detectors, such as electron multipliers, generate signals with "average random deviation" proportional to the square root of signal intensity (23). The proportionality constant was determined empirically (for GC-MS) and was invariant to *m/z* across a variety of MS platforms except at lower signal strengths where background noise becomes dominant. Assuming "average random deviation" refers to mean standard deviation, these results match closely to those of Anderle *et al.* (20).

We would conclude, largely on the basis of the Anderle study (20), that log transformation of MS data is appropriate, and that the error model they presented can be used to properly stabilize the variance of low-intensity peaks, or to obtain robust estimates of variance. It would also be interesting to find out whether these transformations could be applied directly to a raw data matrix rather than to extracted peak abundances.

HIGH-LEVEL PROCESSING

With high-level processing, MS data is harnessed to tackle the ultimate goals of profiling, such as:

1. Sample classification (*e.g.* using proteomic profiles to distinguish chemotherapy-resistant and -sensitive tumors).
2. Low-level biomarker discovery: discerning which data points or peaks are responsible for pattern differences between classes.
3. High-level biomarker discovery: discerning which proteins or peptides correspond to biomarkers found in low-level biomarker discovery.
4. Discerning the full set of proteins present in a sample, and the levels these are expressed at.

These tasks are listed from most-to-least tractable, although the latter two have not been fully addressed in the literature. Building classification algorithms to tease apart class-dependent data is often feasible. Determining the complete set of features (*i.e.* peaks, peptides, or *m/z* scan header tuples) responsible for pattern differences, however, in a statistically sound way, can be difficult. Frequently in machine-learning methods, a feature selection step is used to automatically select features thought to lead to a near-optimal classifier—sometimes independently from learning the classification model (called filtering), or interactively with learning (wrapper), or by way of the classification algorithm itself (*e.g.* decision trees). Feature selection is an open, difficult research question. Even if a theoretically optimal set of features for a particular learning algorithm (in the sense of providing the best generalization) can be found, not all statistically discriminative features will necessarily be contained in this set (or even only such features), because the optimal set of features is heavily dependent on the mathematical framework of the algorithm at hand. For example the optimal set of features in a linear classifier would not contain important pairs of features, which acted only together to provide discriminative power (*e.g.* imagine two proteins, which, if the expression of only one is known, provides no information, but when both are known, they give perfect discrimination). These artifacts, where features optimal for particular classifiers do not necessarily correlate best with the classes, occur with virtually every classifier, offering an explanation to biologists seeking to know why different analyses of the same MS data often shed light on different features, without there being an error or problem in the analysis as has been suggested (9).

STATISTICAL ISSUES

Statisticians take a different view to finding discriminative features. Typically, a test deemed appropriate (*e.g.* a *t* test) is performed, generating a score reflecting how much a feature discriminates between two classes. Then, under the assumption that no features are discriminative, the distribution of these test scores is modeled. From this null distribution, one can calculate the probability (the *p* value) that an individual feature would have a particular test value, or a more extreme one, given that the feature is not discriminative. In the classical setting, these null distributions are theoretically developed, but are now more commonly simulated, for example by permutation tests. An example of a permutation test would be to remove sample labels (*e.g.* cancer and healthy) and then randomly reassign labels so that only spurious correlation exists, with the value of the test function computed for every feature. By repeating this procedure many (*e.g.* 10,000) times, one can obtain an approximation to the null distribution for data with structure similar to that on hand. Permutation tests are attractive because they require no assumptions about the data, in contrast to most classical statistical tests.

Typically permutation tests are performed one feature at a time, resulting in thousands of significance tests. In such cases, the false-positive rate (FPR), which *p* value thresholds seek to control, will be extremely large for commonly used scientific *p* values (*e.g.* $\leq$0.01), and therefore some sort of multiple hypothesis testing correction needs to be applied. Many methods have been developed for this, including the conservative Bonferroni factor, which controls the family-wise error rate over all tests. However, as pointed out by Storey and Tibshirani (17), in the context of biological experiments, it is rather the false-discovery rate (FDR) one seeks to control. Whereas the FPR predicts how many truly null features are expected to be called significant, the FDR predicts how many of the features called significant are in fact likely not to be. From a practical standpoint, the FDR tells us how many "hits" (*e.g.*

peaks putatively different between two classes) are likely false.

Storey and Tibshirani recently developed the *q* value, which is similar to the *p* value, except that it controls for the FDR (rather than FPR) and automatically accounts for multiple testing (17). If one chooses a *q* value of 0.05, then one can expect that 5% of all features deemed significant will in fact not be. The *q* value is therefore a more appropriate statistical measure for biomarker discovery in the context of complex, multivariate datasets.

Permutation tests are sometimes used to test the validity of a classification algorithm (28, 39); the classifier is retrained using randomly permuted labels (with the number of samples in each class maintained) and its predictive power assessed. Such an analysis provides similar information to that obtained from cross-validation, but in manner better suited to reporting the *p* values typically seen in the biomedical literature (as opposed to the machine-learning literature, which does not commonly use *p* values).

Whether seeking discriminative features or aiming to build a classifier, the type of features must be decided upon (*e.g.* intensity values in time, *m/z* space, or peak abundance). If not using peaks, then feature correlation, due for example to isotope shoulders, system imperfection blurring *m/z* values, and the fact that peptides do not elute instantaneously, should be carefully considered. Thus it might be preferable to use quantified peak abundances, combining ions belonging to a single peptide, because these have a clear, intuitive biological meaning. In a discussion concerning the validity of classification using *m/z* values in a set of prostrate cancer proteomic studies with different results, it was suggested that only features with meaning, such as peaks, should be used in biomarker discovery (36). Tibshirani *et al.* advocate the same approach (22). However, it may be wiser to first perform classification, and then look for peak evidence *post hoc* because imperfect peak detection can cause important information to be missed, and because use of peaks over continuous signal does not in fact afford any additional statistical rigor. Assuming the proper methodology is used to validate the classification algorithm, for example by cross-validation, spurious signal should not be identified as a point of interest, regardless of the features used.

MACHINE LEARNING AS APPLIED TO MS

Machine-learning techniques seek to semi-automatically build and validate mathematical models of data that can then be used for classification or regression and for examining which parts of the data were relevant and in what way. A glossary of a few common machine learning terms is provided in Table II. Application of machine-learning techniques to a dataset involves four steps: 1) positing a class of mathematical or statistical models appropriate for the data (the class to choose is not known *a priori*—each model class can work well or perform poorly in different contexts); 2) "learning" which particular model in the class is most suitable for the data (this typically involves a numerical optimization of some objective function to produce a fixed set of parameters identifying a specific model within the model class); and 3) validation of the model by use of a test set, cross-validation, or similar method. Then the fourth and final step can be performed: 4) application of the final model to new data.

A classic MS prediction paper is the pioneering ovarian cancer study of Petricoin *et al.* (8) in which a genetic algorithm was used to find a good set of predictive SELDI *m/z* values. The general idea was to evaluate feature subsets, each 20 *m/z* values (out of ~15,200 total), for the ability to discriminate between cancer and healthy patient serum samples (evaluated based on the ability of the selected features to form two clusters with correct class membership), and then to combine and change the best sets until perfect discriminating power was obtained on the training data. The final model was built by clustering the data with the chosen features, and classifying new samples based on proximity to the class clusters. Because the final optimal set of features appears to be derived from "fitter" subsets formed from new groups of *m/z* rather than newly derived features, the final classifier is a linear classifier in a subset of the original input space. As such, it can be argued that it makes more sense to use a regularized linear discriminant classifier such as nearest shrunken centroid (22, 50), considered by some to be more principled than a genetic algorithm. After training, a holdout test set was evaluated providing 100% specificity and 95% sensitivity. Baggerly *et al.* applied a similar approach (28).

The Petricoin study has sparked an ongoing debate due to the fact that the putative biomarkers discovered were not conclusively proven to be proteins of interest, with those arguing for the need for identification prior to use as a diagnostic and others suggesting that discriminating patterns alone are sufficient (11, 36, 51). The ability to obtain near perfect classification has since been widely reported in the MS literature (8, 28, 48, 52, 53), but reproducibility of results has been limited. There is speculation that improper experimental methodology is to blame, and that the patterns observed are not in fact related to disease state, but instead to extraneous factors such as sample collection, preparation, or storage (9, 11, 22, 36, 53). Thus, a major immediate goal is not so much to develop new classification and biomarker discovery algorithms, but instead to ensure that the datasets in use are in fact representative of the biological problems one seeks to address (an issue beyond the scope of this review).

Here, we briefly report on the major types of methods used to date. We note, however, that it is difficult to empirically judge these techniques as different sample sizes, error estimates (*e.g.* bootstrap, *n*-fold cross-validation with varying *n*, *etc.*), and MS platforms were used without recourse to a reference "gold standard" dataset, and there is a paucity of follow-up biological validation experiments.

Yasui *et al.* used boosted, single-feature, linear regression, and an early stopping criterion based on sensitivity and spec-

TABLE II
Glossary of useful statistical and machine-learning terms

1 Generative Model—A probabilistic model in which observed samples can be viewed as having been generated by a sequence of events described by the model.

2 Hidden Markov Model (HMM)—A type of graphical model (see below) used to model sequence data (*e.g.* amino acid sequences, continuous MS signal). The data is modeled as though it were generated by traversing through a sequence of "hidden" states (hidden because they are unknown to the user). In each hidden state visited, a "symbol" (*e.g.*, an amino acid) is emitted. Efficient algorithms have been developed for HMMs that allow computations to be performed quickly.

3 Graphical Model—A formalism used to efficiently model joint probabilities (*e.g.*, the joint probability over all measured protein levels, cancer aggressiveness, age, and sex). Graphical models can be used for probabilistic supervised or unsupervised machine learning, and naturally accommodate missing values, as well as latent variables.

4 Latent Variable—A variable in a model which is never directly observed. For example, the cluster number in a clustering algorithm can be viewed as a latent variable. Though we never observe its value, positing its existence makes it easier to make sense of the data. Its value is inferred (usually probabilistically) from the observed data.

5 Learning—The stage of machine learning in which parameters of a model class are fit to the data. Also called training. Learning can be supervised or unsupervised.

6 Supervised Learning—Learning in the case where class labels are known. (*e.g.*, which samples belong to cancerous *versus* healthy people). For example, training a discriminative classifier uses supervised learning.

7 Unsupervised Learning—Learning in the case where class labels are unknown. For example, clustering is unsupervised because no class labels are provided.

8 Classifier—A model of data whos intended use is to make class predictions of unseen data. Sometimes called a discriminant model (because it discriminates between two classes).

9 Generalization—The ability of a learned model to generalize to new data (*i.e.* will it work well on unseen data).

10 Linear Discriminant Analysis (LDA)—A type of classifier in which feature values are linearly combined to obtain a class prediction.

11 Linearly Separable—A two-class data set is said to be linearly separable if in the input space, a line (or in two dimensions, a plane, or in higher dimensions, a "hyper-plane") can perfectly separate the two classes of data.

12 Support Vector Machine (SVM)—A class of models that extend the notion of simple linear classifiers to more complex classifiers by projecting the input data into a user-selected, higher-dimensional space (the space is determined by the choice of "kernel"). Even if the data is not linearly separable in the original input space, it may be separable in the kernel space. SVMs are said to have good generalization bounds because of the principle of "margin maximization" at the core of their theoretical development; this principle states that of all the linear classifiers that can separate the input data, one should choose the one that lies furthest from all of the training points.

13 Decision Tree—A classifier in which a new sample is classified by making a series of (usually binary) decisions, which correspond to a traversal through the nodes of the learned tree. The decision made by the tree amounts to following one of a fixed set of rules, such as: If sex = female AND osteopontin_level = high AND noFamilyHistory THEN Class = 1 (healthy), where each part of the rule corresponds to a node in the tree.

14 Decision Stumps—A decision tree with only one node.

15 Random Forest (RF)—A classifier consisting of many decision trees, each one trained with its own subset of the original data, chosen by sampling from the whole training set (with replacement). Classification is performed by having the trees vote on a new sample.

16 *K*-Nearest Neighbor—One of the simplest possible classifiers in which the model consists of "memorizing" all training cases, and then predicting the class of a new sample to be the majority class of its *k* nearest neighbors.

17 Receiver-Operating Characteristics (ROC) Curve—A plot used to measure the sensitivity/specificity trade-offs for a particular classifier. The area under the ROC curve is a measure of how well a classifier will perform over all possible sensitivities (or specificities).

18 Boosting—A technique in which multiple instances (different parameter settings) of a simple classification model (*e.g.* a decision stump) are used together to provide a much stronger model. Each model instance is weighted by an amount related to how well it performed on a weighted subset of the training data. The data weights are in turn related to how well the previous combined set of models performed collectively on the data.

19 Monte Carlo Methods—A general class of techniques whereby exact solutions are approximated by using randomly generated samples.

20 Dynamic Programming (DP)—A class of algorithms used to solve certain kinds of optimization problems by storing the solution to subsets of the overall problem, and then using properties of how the subset solutions relate to the overall solution in an incremental way.

21 Expectation-Maximization (EM)—An algorithm used to perform numerical optimization in the context of latent variable models. One iterates between estimating the value of the latent variables (in the E-Step), and fitting the model parameters (in the M-Step), until convergence.

22 Dynamic Time Warping (DTW)—An algorithm used to align one sequence to another, using DP.

23 Overfitting—The phenomenon in which a model that has too many free parameters relative to the amount of data, ends up fitting (after training) not only the true signal, but also noise and spurious correlations in the data. A model which has overfit will not make good predictions on new, unseen data (*i.e.*, it will not generalize well).

24 Regularization—A technique whereby constraints are added to the objective function so that the effective number of free parameters is reduced, and thus the capacity for overfitting is likewise reduced. For example, one of the simplest methods would be to force some of the free parameters to have the same value.

25 Objective Function—A function set up for training, and which, during training, is maximized (or minimized, depending on the type of function). For example, a typical objective function might be amount of error produced by a classifier on the training examples (data samples used during training), which would be minimized.

26 Shrinkage (*e.g.* Nearest Shrunken Centroid, NSC)—A type of regularization whereby class-specific centroids are "shrunk" toward the overall (nonclass-specific) centroid. This has the effect of eliminating the influence of the most weakly correlated features, thereby reducing the capacity to overfit.

TABLE II—*continued*

27 Lasso—A regularization method whereby the absolute value of the parameters is included in the objective function so as to pressure the absolute value of the parameters to be small (this has been shown to encourage a reduced, effective number of features).

28 Hierarchical Clustering—A clustering algorithm in which the two closest (as defined by some similarity measure) samples (*e.g.* one sample might be all protein levels measured in serum from one patient) are merged to form a node. Progressively, the next two samples (or nodes, or sample/node) are merged into a node, until all samples are in some node. The sequence in which samples are merged into nodes defines a set of hierarchical clusters. By choosing a minimal level of similarity at which to cut off the merging process, a single clustering of the data can be obtained.

29 Feature Selection—A step in machine-learning methods in which certain features are chosen to be included in a model, while others are chosen to be omitted.

30 Wrapper—A feature selection technique in which features are selected in conjunction (and interactively) with the model being trained. This allows the feature selection process to be tailored to the model at hand.

31 Filter (in the context of feature selection)—A feature selection technique in which features are selected before training of a model, and independently of the model, for example using a *t*-test to rank features and then choosing the highest ranking *N* of them. This is distinct from the meaning of filtering in signal processing where filtering refers to application of a digital filter to the data in order to reduce noise or otherwise change the properties of the signal.

32 Principal Components Analysis (PCA)—A technique commonly used to reduce the dimensionality of data. This is accomplished by finding a new space for the data to lie in, in which the new coordinates are linear combinations of the old ones, and in which the new coordinates (called principal components or PCs) are ordered by the amount of variance for which they account in the original data space. The first PC accounts for the most variation in the original data, and each next PC accounts for progressively less. PCs near the bottom frequently account for almost no variance, and hence can be omitted, producing fewer coordinates, and therefore a lower dimensionality. If one has a point in PCA space, one can “back-project” it to see what point it corresponds to in the original space.

33 Unfolded PCA—In standard PCA, each sample is a vector of values. If one has a matrix of values (*e.g.* in the case of one LC-MS data), one can “unfold” the matrix into a vector. This allows standard application of PCA, but throws away some of the information conveyed by storage in a matrix.

34 Bayesian Methods—While many people consider Bayesian methods to be the integration of prior knowledge into the learning task by way of Bayes’ Rule, a more complete description would be centered around the following: Whereas conventional training methods seek to find the single best set of parameters that fit the training data, Bayesian methods seek to represent the uncertainty about the relationships being learned. A practical implication of this, for instance, would be in making predictions from a learned model. In non-Bayesian methods, after learning, one would use a single set of learned parameters to make predictions, whereas in Bayesian prediction, one would integrate out over all possible sets of parameters, weighted by their probabilities (where the probabilities are learned during training) [www.faqs.org/faqs/ai-faq/neural-nets/part3/section-7.html].

35 Cross-validation—A method for making the most use of a dataset for both learning and validation. Rather than separating the data into a single learning set (called the “training” set) and a single test set, *n*-fold cross-validation separates the data into *n* training sets and *n* test sets. If *n* were equal to 5, cross-validation would work as follows: The entire dataset would be divided into five equal-sized groups. The first four groups would be used as training data, and the fifth as test data. The second through to fifth groups would then be used as training data and the first group as test data. This procedure is continued until each group has been used as test data. The aggregate test results from all $n = 5$ phases of the cross-validation would be used to obtain a final estimate of the predictive accuracy. Cross-validation provides an estimate of how a particular model might do on a new, unseen data set drawn from the same statistical distribution.

36 Bootstrap—A technique to assess the variability of a classifier’s predictive ability (or other statistical quantities) by repeatedly measuring the classifier’s predictive ability, using a different subset of the data (chosen by sampling uniformly from the original data set, with replacement) each time.

37 Wavelet Decomposition—A transformation used in signal processing that breaks down a signal into both 1) local features (*i.e.*, spanning only a few adjacent *m/z* values) and 2) frequencies.

38 *F* Statistic—A test statistic sometimes used to rank features for feature selection. It measures the ratio of between-class variance to within-class variance. Higher *F* statistics indicate more class-informative features.

39 Coefficient of Variation (CV)—Defined to be the variance divided by the mean. This is useful if one is measuring how variable two different features are when measured on different scales.

40 Mahalanobis Distance—A distance measure (as opposed to say a Euclidean distance), in which the variation (*e.g.* the noise or scale) of each variable is taken into account, as well as the correlation between variables.

41 Genetic Algorithm—An algorithm in which many features subsets are chosen at random, and their predictive power assessed (in the context of some model, for example, a linear discriminant). In a process of “natural selection,” those feature subsets deemed good (or “fit”) are “mutated” and “crossed” (*i.e.* various operations are performed that change and combine good feature subsets) to produce new feature subsets. This process is continued iteratively until some preset threshold, for example the predictive power on the training set, is met.

42 Fitness Function—The function that measures the “goodness” of particular feature subsets in the context of genetic algorithms.

ificity in their SELDI-MS training set (31). From an initial pool of ~12,000 processed *m/z* values, the final classifier used 25 features (binarized *m/z* intensities).

Qu *et al.* applied boosted decision stumps on SELDI data to obtain near-perfect classification results, with feature selection (*m/z* peaks) based on receiver-operator characteristic (ROC) performance, although the type of classifier used to generate the ROC curve is not stated (52).

Satten *et al.* used a random forest (RF) algorithm to classify MALDI *m/z* values, while Lilien *et al.* used PCA followed by linear discriminant analysis (LDA) on SELDI data, computing feature importance by back-projecting the PCA hyperplane onto the *m/z* space (3, 48).

Wagner *et al.* used each of: *k*-nearest-neighbor ($k = 6$, Mahalanobis distance), support vector machine (SVM) with linear kernel, LDA, and quadratic discriminant analysis (QDA) to classify MALDI data (39), selecting the top 3–15 peaks as features with an *F* statistic. They correctly state that their evaluation is flawed in that some of the methods use the entire dataset for training (*e.g.* to calculate the covariance matrix for the Mahalanobis distance), even though this was not necessary.

Li *et al.* selected 10 *m/z* values as features in three SELDI datasets trying both a *t*-test filter and a genetic algorithm. These were used in conjunction with an SVM classifier, where the choice of kernel was reported to have little effect (settling on a linear one) (54). Overall, the genetic algorithm performed better than the *t*-test approach, suggesting that higher-order interactions between features provide discriminative power, or that the *t*-test is a less-ideal statistic in this scenario.

Wu *et al.* (47) compared the performance of LDA, QDA, *k*-nearest neighbor ($k = 1$–3, Euclidean metric), bagging and boosting classification trees, SVM (kernel not specified), and RF on MALDI data, using both a *t*-test rank and the by-product of the RF algorithm for *m/z* feature selection (15 and 25 features). Overall, no substantive differences in performance were reported, with QDA marginally best, although different error estimators (cross-validation or bootstrap) were used for different classifiers, complicating interpretation.

Baggerly *et al.* undertook an unusual feature selection approach (28). After computation of over 60,000 *t*-tests on individual MALDI *m/z* values, many of the most discriminative *m/z* values appeared as nonpeaks (*i.e.* on the slope of a peak). Dissatisfied with this, they reduce dimensionality by binning *m/z* (bins growing smoothly in size with increasing *m/z*), retaining only those bins (~1%) with detected signal in some minimal number of samples. They note that this binning partially corrects for the correlation between neighboring *m/z* values.

Tibshirani *et al.* set out to develop a classification algorithm, peak probability contrasts (PPC), for MALDI/SELDI data that provides a measure of discriminatory power for all features using simple peak information (22). First, split-points for each peak were found that most discriminate between the two classes. Next, binary features were formed from the split-points and then nearest shrunken centroid (NSC) was applied, with importance assigned to each feature by: i) defining a test statistic based on the difference in the shrunken class proportions at each peak site; ii) using permutation tests to estimate the FDR for different thresholds on the test statistic; and iii) choosing a test-statistic threshold that provided a low FDR (*e.g.* 5%). Using an artificial spike-in experiment, a non-shrunken centroid method reportedly identified spiked features better than the shrunken model, but at the expense of more false-positives and worse classification. Lastly, they compared their PPC algorithm to: i) LDA, using the same 15 features as PPC; ii) SVM (kernel not mentioned, using the same 15 features from PPC and also all *m/z*; iii) binary peak probability features with the Lasso[3]; iv) PPC, but using only peak presence/absence as a feature (not using split-points); and v) using six wavelet coefficients (spanning all *m/z* values) as features with NSC. All-feature SVM performed slightly better than standard PPC, which performed similarly to the Lasso, while LDA, 15-feature SVM, the wavelet method, and presence/absence PPC all performed worse. Note that although the PPC classifier operates on binary data, it uses intensity information by way of split-points (but not fully because split-point distance was not used). Better results might be obtained if this information were incorporated.

Classification in the context of LC-MS data has been more limited, with only a handful of articles (18, 34, 35). After resolving and aligning LC-MS data matrix components (see above), Idborg *et al.* used PCA, unfolded-PCA, partial least squares (PLS), and PARAFAC to tease apart the proteomic differences in the urine of normal mice and drug-administered mice (34). The dimensionality of the classification problem was kept reasonably small and tractable using the abundance of each spectral component at each retention time rather than the full data matrix. In contrast, Radulovic *et al.* performed classification by simple plotting and visualization (18). Evaluation of this approach was limited to only two training cases per class and a single test sample (which can always be perfectly classified by choosing the appropriate point on an ROC curve). A comment was made that if two classes do not sufficiently cluster in the input space, then virtually all classification algorithms will fail. But any nonlinear method can potentially overcome this problem. Moreover, the key motivation for using kernel methods (*e.g.* SVMs) is the fact that even if two classes are not linearly separable in the input space, then one can project data vectors into a higher-dimensional space into which they may become separable. Thus, inability to perform classification in the original input space (or subset of it) should not stop one from further exploration of classification algorithms.

It is useful to ask how differences in the data generated by chromatographic *versus* nonchromatographic MS affect classification and, in particular, if the preprocessing methods reported for LC-MS data are good enough to allow comparable classification results to those reported for MALDI/SELDI (keeping in mind the fact that different peptides are often detected by these platforms). The ability to perform effective time alignments for LC-MS is far more crucial than the ability to align MALDI/SELDI *m/z* values, because the latter problem is less severe. If peak detection is not used prior to LC-MS classification, then correlation across features needs to be taken into account, especially in the time domain where peaks are almost never instantaneous. Aside from these issues, development and use of classification algorithms need not differ between LC-MS and MALDI/SELDI.

An area not studied in any great detail is the use of less-greedy feature selection. In the context of MS, most of the algorithms cited above rank features on an individual basis, using for example a *t*-test, and then greedily take the top *n*

[3] A linear regression model with L1 norm penalty on the regression coefficients.

features. This completely ignores interaction between features. While genetic algorithms are less greedy, many other heuristics can be used to achieve a similar (or possibly better) end and may be worth exploring.

EVALUATING CURRENT METHODS

We have described statistical and computational issues and methods for MS proteomic profiling, without discussing how to evaluate the merits of various techniques—an important and difficult problem. In some studies, the CV is reported as a measure of experimental quality, from start to peak quantification. Several groups have reported CVs ranging from 25 to 60% (the latter, before processing) (10, 18, 31). While these values can be used to assess the contributions of processing steps (for example, by comparing CVs before and after alignment/normalization), they are not absolute measures of quality and certainly much less informative than in the case of microarray data, where the number of features is determined *a priori* by the chip. With MS, feature detection is part of the analytical process, which makes interpretation of CVs drastically different. Consider cases where one uses conservative preprocessing steps, for example heavily smoothing and then binarizing data, retaining only strong signal. In such scenarios, the CV would only reflect dominant peaks surviving preprocessing, which could have much lower CVs than CVs resulting from another processing technique preserving more MS information. Ultimately, what we are interested in is performance on some final goal, such as sample classification. If goal performance is measurable, as in the case of classification, then such measures can provide a more objective assessment of processing steps, as for example is done by Listgarten *et al.* for microarray data to assess normalization schemes (55). Clearly, a feedback loop with the experimental laboratory is also desirable.

FUTURE PROSPECTS

While the application of computational and statistical methods to proteomic profiling is relatively new, it is rapidly gaining interest. Hence, it is worthwhile suggesting fruitful avenues for moving forward. It was suggested above that simultaneous LC-MS data alignment and normalization may be beneficial for comparative profiling. Pushing this further, one could attempt to combine as many of the processing steps as possible into a single model. For example, the probabilistic, generative CPM (19), which takes into account the normalization and alignment problems together, could be extended to also model additive baseline patterns, as well as the error model reported by Anderle *et al.* (20). By combining these steps into a single, probabilistic model, these tasks could be performed simultaneously with one another, across all samples. This is important for two reasons. First, the interplay between various processing steps is unclear, and the order of operation could affect the results. Solving these simultaneously avoids this problem. Second, shared, simultaneous processing across multiple samples allows access to their common structure, and therefore more information. Satten *et al.* emphasize that their methods operate one sample at time (48), so that test data can be treated independently. But it is also possible to process all training data together, and test samples independently, allowing for maximal use of the available data. For example, with the CPM, training data could be combined for normalization and alignment, and then a single test sample brought in and aligned to the model.

One can imagine building a classifier directly on top of the CPM, in a fairly straightforward manner for SELDI/MALDI data. For example, after extending the CPM to non-replicate data, a latent trace could be obtained for each class. The simplest classifier would involve finding the likelihood of a sample for each of the class-specific latent traces and then assigning the class label of the trace producing the highest likelihood. Exploring differences between the latent traces could also be used for biomarker discovery, and prior information of expected differences (*e.g.* sparsity of differences incorporated). A model of this type would provide a unified framework in which preprocessing, classification, and biomarker discovery are systematically tackled, possibly providing better results than sequential, *ad hoc* approaches. We anticipate that existing and emerging statistical and computational techniques, side-by-side with rigorous and systematic evaluation, will help to unleash the full biomedical potential of proteomic profiling.

‖ To whom correspondence should be addressed: CH Best Institute, 112 College St., Toronto, Ontario M5G 1L6, Canada. Tel.: 416-946-7281; Fax: 416-978-8528; E-mail: andrew.emili@utoronto.ca.

REFERENCES

1. Kearney, P., and Thibault, P. (2003) Bioinformatics meets proteomics—Bridging the gap between mass spectrometry data analysis and cell biology. *J. Bioinform. Comp. Biol.* **1,** 183–200
2. Tyers, M., and Mann, M. (2003) From genomics to proteomics. *Nat. Rev. Genet.* **422,** 193–197
3. Lilien, R. H., Farid, H., and Donald, B. R. (2003) Probabilistic disease classification of expression-dependent proteomic data from mass spectrometry of human serum. *J. Comput. Biol.* **10,** 925–946
4. Cartegni, L., Chew, S. L., and Krainer, A. R. (2002) Listening to silence and understanding nonsense: Exonic mutations that affect splicing. *Nat. Rev. Genet.* **3,** 285–298
5. Lewin, B. (2004) *Genes VIII*. Prentice Hall, Upper Saddle River, NJ
6. Aebersold, R., and Mann, M. (2003) Mass spectrometry-based proteomics. *Nature* **422,** 198–207
7. Kislinger, T., and Emili, A. (2003) Going global: Protein expression profiling using shotgun mass spectrometry. *Curr. Opin. Mol. Ther.* **5,** 285–293
8. Petricoin, E. F., Ardekani, A. M., Hitt, B. A., Levine, P. J., Fusaro, V. A., Steinberg, S. M., Mills, G. B., Simone, C., Fishman, D. A., Kohn, E. C., and Liotta, L. A. (2002) Use of proteomic patterns in serum to identify ovarian cancer. *Lancet* **359,** 572–577
9. Diamandis, E. P. (2004) Mass spectrometry as a diagnostic and a cancer biomarker discovery tool: Opportunities and potential limitations. *Mol. Cell. Proteomics* **3,** 367–378
10. Wang, W., Zhou, H., Lin, H., Roy, S., Shaler, T. A., Hill, L. R., Norton, S., Kumar, P., Anderle, M., and Becker, C. H. (2003) Quantification of proteins and metabolites by mass spectrometry without isotopic labeling or spiked standards. *Anal. Chem.* **75,** 4818–4826
11. Petricoin, E. F., and Liotta, L. A. (2003) Counterpoint: The vision for a new diagnostic paradigm. *Clin. Chem.* **49,** 1276–1278

12. Petricoin, E. F., and Liotta, L. A. (2003) Mass spectrometry-based diagnostics: The upcoming revolution in disease detection. *Clin. Chem.* **49,** 533–534
13. Hastie, T., Tibshirani, R., and Friedman, J. (2001) *The Elements of Statistical Learning*. Springer, New York
14. Deleted in proof
15. Liu, H., Sadygov, R. G., and Yates, J. R., 3rd (2004) A model for random sampling and estimation of relative protein abundance in shotgun proteomics. *Anal. Chem.* **76,** 4193–4201
16. Oppenheim, A. V., Willsky, A., and Nawab, S. (1996) *Signals and Systems*. Prentice Hall, Upper Saddle River, NJ
17. Storey, J. D., and Tibshirani, R. (2003) Statistical significance for genomewide studies. *Proc. Natl. Acad. Sci. U. S. A.* **100,** 9440–9445
18. Radulovic, D., Jelveh, S., Ryu, S., Hamilton, T. G., Foss, E., Mao, Y., and Emili, A. (2004) Informatics platform for global proteomic profiling and biomarker discovery using liquid-chromatography-tandem mass spectrometry. *Mol. Cell. Proteomics* **3,** 984–997
19. Listgarten, J., Neal, R. M., Roweis, S. T., and Emili, A. (2005) Multiple alignment of continuous time series. In *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, MA
20. Anderle, M., Roy, S., Lin, H., Becker, C., and Joho, K. (2004) Quantifying reproducibility for differential proteomics: noise analysis for protein liquid chromatography-mass spectrometry of human serum. *Bioinformatics* **20,** 3575–3582
21. Petricoin, E. F., and Liotta, L. A. (2004) SELDI-TOF-based serum proteomic pattern diagnostics for early detection of cancer. *Curr. Opin. Biotechnol.* **15,** 24–30
22. Tibshirani, R., Hastie, T., Narasimhan, B., Soltys, S., Shi, G., Koong, A., and Le, Q. T. (2004) Sample classification from protein mass spectrometry, by peak probability contrasts. *Bioinformatics* **20,** 3034–3044
23. Stein, S. E. (1999) An integrated method for spectrum extraction and compound identification from GC/MS data. *J. Am. Soc. Mass Spectrom.,* **10,** 770–781
24. Windig, W., Phalp, J. M., and Payne, A. W. (1996) A noise and background reduction method for component detection in liquid chromatography/mass spectrometry. *Anal. Chem.* **68,** 3602–3606
25. Bylund, D. (2001) *Chemometric Tools for Enhanced Performance in Liquid Chromatography-Mass Spectrometry*. Uppsala University, Uppsala, Sweden
26. The MathWorks Inc. (2004) *Curve Fitting Toolbox User's Guide, Revised for Version 1*. The MathWorks Inc., Natick, MA
27. Sauve, A., and Speed, T. (2004) Normalization, baseline correction and alignment of high-throughput mass spectrometry data. *Proceedings Gensips*
28. Baggerly, K. A., Morris, J. S., Wang, J., Gold, D., Xiao, L. C., and Coombes, K. R. (2003) A comprehensive approach to the analysis of matrix-assisted laser desorption/ionization-time of flight proteomics spectra from serum samples. *Proteomics* **3,** 1667–1672
29. Press, W. H., *et al.* (1992) *Numerical Recipes in C: The Art of Scientific Computing*, 2nd Ed., Cambridge University Press, Cambridge, United Kingdom
30. Hastings, C. A., Norton, S. M., and Roy, S. (2002) New algorithms for processing and peak detection in liquid chromatography/mass spectrometry data. *Rapid Commun. Mass Spectrom.* **16,** 462–467
31. Yasui, Y., Pepe, M., Thompson, M. L., Adam, B. L., Wright, G. L., Jr., Qu, Y., Potter, J. D., Winget, M., Thornquist, M., and Feng, Z. (2003) A data-analytic strategy for protein biomarker discovery: Profiling of high-dimensional proteomic data for cancer detection. *Biostatistics* **4,** 449–463
32. Randolph, T. W., and Yasui, Y. (2004) Multiscale processing of mass spectrometry data, UW Biostatistics Working Paper Series, Working Paper 230
33. Chau, F.-T., Liang, Y.-Z., Gaon, J., and Shao, X.-G. (2004) *Chemometrics: From Basics to Wavelet Transform*. Wiley, New York, NY
34. Idborg, H., Edlund, P. O., and Jacobsson, S. P. (2004) Multivariate approaches for efficient detection of potential metabolites from liquid chromatography/mass spectrometry data. *Rapid Commun. Mass Spectrom.* **18,** 944–954
35. Idborg-Bjorkman, H., Edlund, P. O., Kvalheim, O. M., Schuppe-Koistinen, I., and Jacobsson, S. P. (2003) Screening of biomarkers in rat urine using LC/electrospray ionization-ms and two-way data analysis. *Anal. Chem.* **75,** 4784–4792
36. Diamandis, E. P. (2003) Point: Proteomic patterns in biological fluids: Do they represent the future of cancer diagnostics? *Clin. Chem.* **49,** 1272–1275
37. *Tutorial on m/z*, Proteometrics LLC, http://bioinformatics.genomicsolutions.com/moverz/tutorials/
38. Deleted in proof
39. Wagner, M., Naik, D., and Pothen, A. (2003) Protocols for disease classification from mass spectrometry data. *Proteomics* **3,** 1692–1698
40. Nielsen, N.-P. V., Carstensen, J. M., and Smedsgaard, J. (1998) Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping. *J. Chromatogr. A* **805,** 17–35
41. Bylund, D., Danielsson, R., Malmquist, G., and Markides, K. E. (2002) Chromatographic alignment by warping and dynamic programming as a pre-processing tool for PARAFAC modelling of liquid chromatography mass spectrometry data. *J. Chromatogr. A* **961,** 237–244
42. Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, United Kingdom
43. Poritz, L. B. (1988) Hidden Markov models: A guided tour. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Morgan Kaufmann, New York
44. Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc. B* **39,** 1–38
45. Sakoe, H., and Chiba, S. (1978) Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust.* **26,** 43–49
46. Quackenbush, J. (2002) Microarray data normalization and transformation. *Nat. Genet.* **32,** (suppl.) 496–501
47. Wu, B., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., Ward, D., Williams, K., and Zhao, H. (2003) Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. *Bioinformatics* **19,** 1636–1643
48. Satten, G. A., Datta, S., Moura, H., Woolfitt, A. R., Carvalho Mda, G., Carlone, G. M., De, B. K., Pavlopoulos, A., and Barr, J. R. (2004) Standardization and denoising algorithms for mass spectra to classify whole-organism bacterial specimens. *Bioinformatics* **20,** 3128–3136
49. Molla, M., Waddell, M., Page, D., and Shavlik, J. (2004) Using machine learning to design and interpret gene-expression microarrays. *AI Magazine.* **25,** 37
50. Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Natl. Acad Sci. U. S. A.* **99,** 6567–6572
51. Diamandis, E. P. (2003) Correspondence: Re: Serum proteomic patterns for detection of prostate cancer. *J. Natl. Cancer Inst.* **95,** 489–491
52. Qu, Y., Adam, B. L., Yasui, Y., Ward, M. D., Cazares, L. H., Schellhammer, P. F., Feng, Z., Semmes, O. J., Wright, G. L., Jr. (2002) Boosted decision tree analysis of surface-enhanced laser desorption/ionization mass spectral serum profiles discriminates prostate cancer from noncancer patients. *Clin. Chem.* **48,** 1835–1843
53. Sorace, J. M., and Zhan, M. (2003) A data review and re-assessment of ovarian cancer serum proteomic profiling. *BMC Bioinformatics* **4,** 24
54. Li, L., Tang, H., Wu, Z., Gong, J., Gruidl, M., Zou, J., Tockman, M., and Clark, R. A. (2004) Data mining techniques for cancer detection using serum proteomic profiling. *Artif. Intell. Med.* **32,** 71–83
55. Listgarten, J., Graham, K., Damaraju, S., Cass, C., Mackey, J., and Zanke, B. (2003) Clinically validated benchmarking of normalisation techniques for two-colour oligonucleotide spotted microarray slides. *Appl. Bioinform.* **2,** 219–228