Predicting Protein Levels from Tandem Mass Spectrometry Data

Anthony Bonner and Han Liu
Department of Computer Science, University of Toronto
{bonner, hanliu}@cs.toronto.edu

***Abstract*— This paper addresses a central problem of Proteomics: estimating the amounts of each of the thousands of proteins in a cell culture or tissue sample. Although laboratory methods have been developed for this problem, we seek a simple method, one that does not involve intricate, complex or expensive laboratory procedures. Instead, our aim is to use machine-learning techniques to infer protein levels from the relatively cheap and abundant data available from high-throughput tandem mass spectrometry (MS/MS). In this paper, we develop and evaluate several techniques for tackling this problem. Specifically, we develop three generative models of MS/MS data, and for each, we develop a family of methods for efficiently fitting the model to data. We prove that each method is correct in that it achieves a well-defined optimization criterion. In addition, to evaluate their biological relevance, we test each method on three real-world datasets generated by MS/MS experiments performed on various tissue samples taken from Mouse.**

I. Introduction

Proteomics is the large-scale study of the thousands of proteins in a cell [7]. In a typical Proteomics experiment, the goal might be to compare the proteins present in a certain tissue under different conditions. For instance, a biologist might want to study cancer by comparing the proteins in a cancerous liver to the proteins in a healthy liver. Modern mass spectrometry makes this possible by enabling the identification of thousands of proteins in a complex mixture [9], [4]. However, *identifying* proteins is only part of the story. It is also important to *quantify* them, that is, to estimate how much of each protein is present in a cell [1], [6]. To this end, a number of laboratory methods have been developed, notably those based on mass tagging with isotopes [5], [8]. However, recent research [10] suggests that simpler, more-direct methods may be possible, methods that do not require complex laboratory procedures, but which are simply based on the spectral counts provided by tandem mass spectrometers. This paper is an initial exploration of this possibility. In particular, we investigate the possibility of using machine learning techniques to infer protein quantity from tandem mass spectrometry data.

Tandem mass spectrometry involves several phases in which proteins are broken up and the pieces separated by mass [7], [9]. First, since proteins themselves are too large to deal with, the thousands of unknown proteins in a cell culture or tissue sample are fragmented into tens of thousands of peptides. The peptides are then ionized and passed through a mass spectrometer. This produces a mass spectrum in which each spectral peak corresponds to a peptide. From this spectrum, individual peptides are selected for further analysis. Each such peptide is further fragmented and passed through a second mass spectrometer, to produce a so-called tandem mass spectrum. The result is a collection of tandem mass spectra, each corresponding to a peptide. Each tandem mass spectrum acts as a kind of fingerprint, identifying the peptide from which it came. By searching a database of proteins, it is possible to identify the protein that produced the peptide that produced the tandem mass spectrum. In this way, the proteins in the original tissue sample are identified. The entire process is completely automatic.

A peptide mixture is not analyzed all at once. Instead, to increase sensitivity, the peptides are "smeared out" over time (often using liquid chromatography), so that different kinds of peptides enter the mass spectrometer at different times. A typical MS/MS experiment may last many hours, with proteins and peptides being identified each second. Copies of a particular peptide may continue to enter the mass spectrometer for several seconds or minutes. As the copies enter, the peptide will be repeatedly identified, once a second. In this way, a peptide may be identified and re-identified many times, increasing the confidence that the identification is correct. Each identification of a peptide is called a *spectral count*, since it requires the generation of a tandem mass spectrum. A large spectral count indicates that a peptide has been confidently identified.

Recent research has shown that the spectral counts of peptides are linearly related to protein abundance. In particular, as the relative abundance of a given protein is increased, the total spectral count of its peptides increases in direct proportion [10]. In effect, more input leads to proportionately more output. However, the relationship is not at all straightforward, since two proteins with the same spectral counts may have different abundances. Thus, despite the linear relationship, different proteins have different proportionality constants. Moreover, at present, there is no way to predict what these constants are. That is, there is no complete quantitative theory relating a protein's abundance to its spectral count.

This paper uses machine-learning techniques to take a first step towards developing such a theory. Specifically, we develop three generative models of MS/MS data, and for each, we develop a family of methods for efficiently fitting the model to data. Because this is an initial study, the models were chosen for their simplicity and tractability, and the goal is to see how well (or poorly) they fit the data, and to quantify the error. Each model predicts the spectral count of a peptide based on two factors: its amino-acid sequence, and the abundance of the protein from which it was derived. The three models differ in their treatment of peptide ionization. However, they each provides an explanation for the linear relationship between protein abundance and spectral count. More importantly, we show how to use each model to estimate protein abundance from spectral count.

To evaluate the models, the Emili Laboratory at the Banting and Best Department of Medical Research at the University of Toronto has provided us with several datasets of several thousand proteins and peptides. The datasets were derived from MS/MS experiments on protein mixtures extracted from various tissue samples of Mouse. A small sample of the data is shown in Table I. (Details on how this data was generated can be found in [11].) Each row in the table represents a peptide. The first (left-most) column is the Swissprot accession number identifying a protein. The second column is the amino-acid sequence of the peptide. The third column is the spectral count of the peptide, and the last column is its charge. (All peptides are ionized as they pass through the mass spectrometer and thus have a charge.) Notice that there may be many entries for the same protein, since a single protein can produce many peptides.

TABLE I
A FRAGMENT OF A DATA FILE

Protein ID	Peptide	Count	Charge
Q91VA7	$TRHNNLVIIR$	4	2
Q91VA7	$KLDLFAVHVK$	3	2
...	...	...	...

High-throughput MS/MS experiments can provide a large amount of data on which to train and test machine-learning methods. However, they also introduce a complication, since the amount of protein input to the mass spectrometer is unknown. This can be seen in Table I, where spectral count is provided, but protein abundance is not. Thus, it is in general unclear whether a high spectral count for a peptide is due to the properties of the peptide or to a large amount of protein at the input. One of the challenges is to untangle these two influences. What makes the problem approachable is that we have data on spectral counts for peptides from the *same protein*, so differences in their counts cannot be due to differences in protein abundance. The models and methods we develop were chosen, in part, because of their ability to exploit this information. In effect, they treat protein abundance as a latent, or hidden variable, whose value must be estimated. In addition, they lead to efficient algorithms based on well-developed operators of linear algebra (specifically, matrix inversion and eigenvector decomposition). We show that the algorithms are correct in that they each achieve a well-defined optimization criterion.

We evaluated our methods and models on real and simulated datasets. While real-world data tests their biological relevance, simulated data tests their mathematical and computational correctness. The tests on simulated data act as a sanity check, since errors in either the mathematics or the programming of the methods can easily appear as unexpected or bizarre results. In addition, since protein levels for the simulated data are known, these tests show that our methods can in principle estimate protein abundance, untangling its influence from that of other factors. To evaluate the methods on real data, we use ten-fold cross validation, with correlation coefficient used to measure the goodness-of-fit of a learned model to the testing portion of the data. The main difficulty is the distribution of the data. As shown in the full paper, the data ranges over several orders of magnitude and is highly skewed, with most data concentrated at very low values. In fact, we show that it has an $O(1/y^2)$ distribution, where y denotes spectral count. To deal with this difficulty, we use the Spearman rank correlation coefficient to measure the goodness-of-fit [2]. Unlike the more common Pearson correlation coefficient, which measures *linear* correlation, Spearman's coefficient measures *monotone* correlation and is insensitive to extreme data values. In addition, we use log-log plots of observed v.s. estimated values to provide an informative visualization of the fit. Finally, we compare our methods to a number of naive methods, and report on their performance.

The full version of this paper is organized as follows. Sections 1 and 2 introduce the biological problem and provide biological background. Section 3 develops our three generative models of MS/MS data. Section 4 develops a number of computational methods for fitting these models to data. Section 5 describes several datasets and an experimental methodology for training and testing our methods. Section 6 uses the datasets to test and compare our methods. Finally, Section 7 summarizes the results and suggests possible extensions for future work.

A draft of the full paper is available on the web [3].

II. MODELING SPECTRAL COUNTS

This section presents our three models of MS/MS data. Each model represents a different hypothesis about the way MS/MS is generated. The main difference between them is their treatment of peptide ionization. The full paper evaluates the models on real MS/MS data and quantifies the error.

To keep track of different proteins and peptides, we use two sets of indices, usually i for proteins and j for peptides. Proteins are numbered from 1 to N, and the peptides for the i^{th} protein are numbered from 1 to n_i. In addition, we use y to denote spectral count, and in to denote the amount of protein input to the mass spectrometer. Thus, in_i is the amount of protein i, and y_{ij} is the spectral count of peptide j of protein i. With this notation, the following equation provides a common framework for our models:

$$y_{ij} \;=\; in_i \cdot ie_{ij} \tag{1}$$

This equation divides spectral count into two factors: in_i, the amount of protein from which peptide ij was generated; and ie_{ij}, the *ionization efficiency* of the peptide. Ionization efficiency can be thought of as the propensity of the peptide to ionize and contribute to a mass spectrum, though it includes all factors that contribute to spectral counts *other than* the amount of protein. In this way, we hope to untangle the amount of protein (which we want to estimate) from all other factors. Note that y_{ij} is observed, while in_i and ie_{ij} are both unknown. Estimating (learning) values for these unknowns is the main goal of this research.

As mentioned earlier, recent research has shown that the abundance of a protein is directly proportional to the total

spectral count of its peptides [10]. That is,

$$in_i \;=\; b_i \sum_j y_{ij}$$

where b_i is an (unknown) proportionality constant. The notion of ionization efficiency provides an explanation for this proportionality and a way of computing the constants b_i. In particular, it follows immediately from Equation 1 that

$$in_i \;=\; \frac{\sum_j y_{ij}}{\sum_j ie_{ij}} \qquad (2)$$

In other words, $b_i = 1/\sum_j ie_{ij}$. Thus, according to the framework of Equation 1, learning ionization efficiencies, ie_{ij}, is the central problem in estimating protein abundance.

It should be noted that with the model and data described above, we can only learn *relative* values of these unknowns, not absolute values. This is because any solution to Equation 1 is only unique up to a constant: multiplying all the in_i by a constant, and dividing all the ie_{ij} by the same constant gives another, equally good solution. However, estimating the relative amounts of protein is an extremely useful biological result. Moreover, by using a small amount of calibration data, the relative values can all be converted to absolute values.

In order to estimate relative values for these unknowns, we need a model of ionization efficiency. In this paper, we investigate three relatively simple models:

$$\begin{array}{lll} \text{Linear :} & ie_{ij} \;=\; \mathbf{x_{ij}} \bullet \beta \\ \text{Exponential :} & ie_{ij} \;=\; e^{\mathbf{x_{ij}} \bullet \beta} \\ \text{Inverse :} & ie_{ij} \;=\; 1/(\mathbf{x_{ij}} \bullet \beta) \end{array}$$

Here, β is a vector of parameters (to be learned), x_{ij} is a vector of (known) peptide properties, and $\bullet$ denotes the dot product (or inner product) of the two vectors. The peptide properties are all derived from the amino-acid sequence of the peptide. They could include such things as length, mass, amino-acid composition, and estimates of various biochemical properties such as hydrophobicity, chargeability, pH under the experimental conditions, etc. The full paper spells out the specific properties used in this study.

We investigate linear models because they are directly amenable to the techniques of linear algebra. We investigate exponential models because, by taking logs, they become linear. In addition, exponential models have the advantage that the ionization efficiency is guaranteed to be positive. In contrast, the linear model may produce a preponderance of positive values, but it sometimes produces negative values as well, which are meaningless (though very small negative values can be assumed to be zero).

The inverse model has a different motivation. As mentioned earlier, spectral counts have a very skewed distribution of values, ranging over several orders of magnitude, with most of the values concentrated at the very low end of the spectrum. In fact, we show that the distribution is $O(1/y^2)$, where y denotes spectral count. It can be difficult to fit a linear model to data with this kind of distribution, since a small number of very large values tends to dominate the fit. Even if the largest values are removed, the next largest values dominate, ad infinitum. Taking logarithms helps, but even $\log(y)$ has a skewed distribution. However, $1/y$ has a uniform distribution, thus eliminating all skew. This is the motivation for the inverse model: to transform the data to a form that is more manageable. In addition, all the methods we develop for fitting the linear model are easily adapted to fit the inverse model.

Note that for each model, once the parameter β is learned, we can use it to estimate the ionization efficiencies of any peptide, including peptides not in the training set. We can then use Equation 2 to estimate the amount of protein in the input sample, since values are now available for both y_{ij} and ie_{ij}. Our experiments in the full paper illustrate this idea by learning β on a training dataset and then applying it to a test dataset to estimate protein abundance.

III. Fitting the Models to Data

The models described above each require the estimation of a parameter vector, β. For each model, this problem would reduce to multivariate linear regression if the amounts of protein, in_i, were included in the training data. Unfortunately, the training data does *not* include this information. This makes each of the models non-linear in the unknowns. The full paper develops a number of methods for transforming these non-linear models into linear ones and for efficiently fitting them to data. In some cases, the problem still reduces to multivariate linear regression, but in most cases it reduces to generalized eigenvector problems. To give a taste of what is involved, we discuss here the methods for the linear model and develop one of them in detail.

A. Linear Models

We develop three methods for fitting the linear model to experimental data, where each method is meant to improve upon the one before. The first two methods are closely related. They have in common that learning is divided into two phases: the first phase estimates a value for β, and the second phase uses β to help estimate values for the amounts of protein, in_i. The two methods differ in the optimization criteria they use to fit the model to the data. The third model is different from the first two in that it has only one learning phase, in which all parameters are estimated simultaneously. In this way, we hope to get a better fit to the data, since the estimate of β is now affected by how well the estimates of in_i fit the data, something that is impossible in the two-phase approach.

Recall that the linear model is given by equations of the form:

$$y_{ij} \;=\; in_i \cdot (\mathbf{x}_{ij} \bullet \beta) \qquad (3)$$

where the parameter vector β and all the in_i are unknown and must be learned. Of course, these equations are not exact, and provide at best an approximate description of the data. The goal is to see how closely they fit the data, and to estimate values for β and in_i in the process. From the discussion in Section II, we know it is only possible to estimate *relative* values for these quantities. This effectively means we can determine the *direction* of β but not its *magnitude*. In fact, in the absence of calibration data, the magnitude of β is meaningless. For this reason, all the methods for the linear

model impose constraints on the magnitude of β in order to obtain a unique solution. We now develop the first (and simplest) of these methods.

1) LIN1: Two-Phase Learning: This approach factors out the amount of protein, in_i, from the set of Equations 3. The result is a linear eigenvector equation for the parameter vector β, which can be solved using standard eigenvector methods.

From Equations 3, we see that protein i gives rise to the following set of equations, one equation for each peptide:

$$y_{i1} \;=\; in_i \cdot (\mathbf{x}_{i1} \bullet \beta) \quad \cdots \quad y_{in_i} \;=\; in_i \cdot (\mathbf{x}_{in_i} \bullet \beta) \quad (4)$$

Note that the unknown value in_i is the same in each equation. Thus, by dividing each equation by the previous one, we can eliminate this unknown value, leaving the parameter vector β as the only unknown quantity. That is, $y_{ij}/y_{i,j-1} = (\mathbf{x}_{ij} \bullet \beta)/(\mathbf{x}_{i,j-1} \bullet \beta)$, for j from 2 to n_i. Cross multiplying gives $y_{ij}(\mathbf{x}_{i,j-1} \bullet \beta) = y_{i,j-1}(\mathbf{x}_{ij} \bullet \beta)$, and rearranging terms gives the following:[1]

$$\mathbf{z}_{ij} \bullet \beta = 0 \quad where \;\; \mathbf{z}_{ij} \;=\; y_{ij}\mathbf{x}_{i,j-1} - y_{i,j-1}\mathbf{x}_{ij} \quad (5)$$

for j from 2 to n_i, and i from 1 to N. Geometrically, these equations mean that the parameter vector β is orthogonal to each of the derived vectors $\mathbf{z}_{ij}$. Note that this is a constraint on the direction of β but not its magnitude, which is to be expected. Equation 5 is a restatement of Equation 3 with the unknown values in_i removed. Like Equation 3, it is an approximation, and our goal is to see how closely we can fit it to the data.

A simple approach is to choose β so that the values of $\mathbf{z}_{ij} \bullet \beta$ are as close to 0 as possible. That is, we can try to minimize the sum of their squares, $\sum_{i,j}(\mathbf{z}_{ij} \bullet \beta)^2$. Of course, this sum can be trivially minimized to 0 by setting $\beta = 0$. But, as described above, the magnitude of β is meaningless, and only its direction is important. So, without loss of generality, we minimize the sum of squares subject to the constraint that the magnitude of β is 1. To do this, we use the method of Lagrange multipliers. That is, we minimize the following function:

$$F(\beta, \lambda) \;=\; \sum_{i,j}(\mathbf{z}_{ij} \bullet \beta)^2 \;-\; \lambda(\|\beta\|^2 - 1) \quad (6)$$

Taking partial derivatives with respect to β and setting the result to 0, we get the equation

$$\sum_{ij} \mathbf{z}_{ij}(\mathbf{z}_{ij} \bullet \beta) \;=\; \lambda\beta \quad (7)$$

Taking the inner product of both sides with β gives $\sum_{ij}(\mathbf{z}_{ij} \bullet \beta)^2 = \lambda\beta \bullet \beta = \lambda\|\beta\|^2 = \lambda$, where the last equation follows from the constraint $\|\beta\| = 1$. We have therefore derived the following two equations:

$$\sum_{ij} \mathbf{z}_{ij}\mathbf{z}_{ij}^T\, \beta \;=\; \lambda\beta \qquad\qquad \lambda \;=\; \sum_{ij}(\mathbf{z}_{ij} \bullet \beta)^2$$

[1]Of course, we could generate many more equations of this form by cross multiplying all possible pairs of equations from 4, instead of just the successive ones. However, only $n_i - 1$ of the resulting equations would be linearly independent.

where the left equation is just Equation (7) expressed in matrix notation, with all vectors interpreted as column vectors. The left equation says that β is an eigenvector of the matrix $\sum_{ij} \mathbf{z}_{ij}\mathbf{z}_{ij}^T$, and the right equation says that its eigenvalue is just the sum of squares we want to minimize. We should therefore choose the eigenvector with the smallest eigenvalue. This provides an estimate of the parameter vector β, and completes the first phase of learning.

In the second phase, we estimate the ionization efficiency of each peptide, using the equation $ie_{ij} = \mathbf{x}_{ij} \bullet \beta$. Finally, from the ionization efficiencies, we estimate the abundance of each protein, in_i, using Equation 2.

REFERENCES

[1] R. Aebersold and M. Mann. *Mass spectrometry-based proteomics*, Nature 422, pp 198-207, 2003.

[2] P. Bickel and K. Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics*, Holden-Day INC, 1977.

[3] A. Bonner and H. Liu. *Development and Evaluation of Methods for Predicting Protein Levels and Peak Intensities from Tandem Mass Spectrometry Data*. Technical report. Available at www.cs.toronto.edu/~bonner/publications.

[4] J.E. Elias, F.D. Gibbons, O.D. King, F. Roth, and S.P. Gygi. *Intensity-based protein identification by machine learning from a library of tandem mass spectra*, Nature Biotechnology. Volume 22 Number 2, 2004

[5] S.P. Gygi, B. Rist, S.A. Gerber, F. Turecek, M.H. Gelb, and R. Aebersold. *Quantitative analysis of complex protein mixtures using isotope-coded affinity tags*, Nature Biotechnology 17, pp 994–999.

[6] S.P.Gygi and R. Aeberold. *Mass Spectrometry and Proteomics*, Current Opinion in Chemical Biology, 4, pp 489–494, 2000.

[7] D.C. Liebler. *Introduction to Proteomics, tools for the new biology*, Humana Press, NJ, 2002.

[8] S. Ong, B. Blagoev, I. Kratchmarovat, D.B. Kristensen, H. Steen, A. Pandey, and M. Mann. *Stable Isotope Labelling by Amino Acid in Cell Culture, SILAC, as a Simple and Accurate Approach to Expression Proteomics*, Molecular & Cellular Proteomics 1.5 2002.

[9] G. Siuzdak. *The Expanding Role of Mass Spectrometry in Biotechnology*, Mcc Press, 2003.

[10] H. Liu, R.G. Sadygov, and J.R. Yates. *A Model for Random Sampling and Estimation of Relative Protein Abundance in Shotgun Proteomics*, Anal. Chem. 76, pp 4193–4201, 2004.

[11] T. Kislinger, K. Ramin, D. Radulovic, et al. *PRISM, a Generic Large Scale Proteomics Investigation Strategy for Mammals*, Molecular & Cellular Proteomics 2.1 2003.