MODEL-BASED BIOMARKER DETECTION AND SYSTEMATIC ANALYSIS IN TRANSLATIONAL SCIENCE

A Dissertation

by

YOUTING SUN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2012

Major Subject: Electrical Engineering

MODEL-BASED BIOMARKER DETECTION AND SYSTEMATIC ANALYSIS IN TRANSLATIONAL SCIENCE

A Dissertation

by

YOUTING SUN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Co-Chairs of Committee,	Edward R. Dougherty
	Ulisses Braga-Neto
Committee Members,	Aniruddha Datta
	Ivan Ivanov
Head of Department,	Costas Georghiades

May 2012

Major Subject: Electrical Engineering

ABSTRACT

Model-based Biomarker Detection and Systematic Analysis

in Translational Science. (May 2012)

Youting Sun, B.S., Tsinghua University

Co–Chairs of Advisory Committee: Dr. Edward R. Dougherty
Dr. Ulisses Braga-Neto

This dissertation is concerned with the application of mathematical modeling and statistical signal processing into the rapidly expanding fields of proteomics and genomics. The research is guided by a translational goal which drives the problem formalization and experimental design, and leads to optimization, prediction and control of the underlying system. The dissertation is comprised of three interconnected subjects.

In the first part of the dissertation, two Bayesian peptide detection algorithms are proposed to optimize the feature extraction step, which is the most fundamental step in mass spectrometry-based proteomics. The algorithms are designed to tackle data processing challenges that are not satisfactorily addressed by existing methods. In contrast to most existing methods, the proposed algorithms perform deisotoping and deconvolution of mass spectra simultaneously, which enables better identification of weak peptide signals. Unlike greedy template-matching algorithms, the proposed methods have the capability to handle complex spectra where features overlap. The proposed methods achieve better sensitivity and accuracy compared to many popular software packages such as msInspect.

In the second part of the dissertation, we consider modeling and assessing the entire mass spectrometry-based proteomic data analysis pipeline. Different modules

are identified and analyzed, resulting in a framework that captures key factors in system performance. The effects of various model parameters on protein identification rates and quantification errors, differential expression results, and classification performance are examined. The proposed pipeline model can be used to aid experimental design, pinpoint critical bottlenecks, optimize the workflow, and predict biomarker discovery results.

Finally, the same system methodology is extended to analyze the workflow in DNA microarray experiments. A model-based approach is developed to explore the relationship among microarray data properties, missing value imputation, and sample classification in a complicated data analysis pipeline. The situations when it is suitable to apply missing value imputation are identified and recommendations regarding imputation are provided. In addition, a missing value rate-related peaking phenomenon is uncovered.

To My Family

ACKNOWLEDGMENTS

I would like to gratefully acknowledge my advisor, Dr. Edward R. Dougherty, whose constant support creates an ideal environment for my doctoral research. His experienced insights shed light on my intellectual development, career path and beyond.

I am also deeply indebted to my co-adviser, Dr. Ulisses Braga-Neto, without whose sincere care and guidance the work would not have been possible. His wit and passion towards research and his dedication to students inspired me throughout the years.

I would also like to thank other fine scientists on my committee, Dr. Aniruddha Datta and Dr. Ivan Ivanov, for their constructive advice. I am grateful to my collaborator Dr. Jianqiu Zhang for many stimulating and helpful discussions, to my colleagues and friends Dr. Jianping Hua, Dr. Golnaz Vahedi, and Dr. Jian Liu for their generous help.

Finally, I would like to thank my parents for their great vision and guidance, endless love and support.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

Translational science translates multidisciplinary scientific research into clinical practice with the goal of aiding diagnosis, discovering new drugs, developing more effective treatments, and thus improving human health. Translational research may have different meaning to different researchers, but it seems important to almost everyone [1].

There are generally two paths in the practice of translational research. One is data-driven and the other is goal-driven. The former tries to make sense of the data, link the observed phenomenon with scientific explanations, or apply pattern recognition to discover features that can be associated to phenotypes. The latter follows the guidance of a goal, which usually leads to carefully designed experiments and conceptualization of the translational problem [2]. In this dissertation, we adopt the goal-driven approach, which is more of a systems engineering approach.

For "conceptualization", a model-based approach is undoubtedly a beneficial way to go as proved by the successful development and application of many well known methods such as ANOVA for microarray data analysis [3], hypothesis testing for biomarker detection, and controlling false positive identifications via decoy databases in mass spectrometry (MS) based proteomics [4, 5]. The beauty of model-based approach lies in that by mathematical formalization of the problem, key issues and factors can be captured, optimal solutions can be achieved, and the gained insights can be translated into prediction or control of the underlying system. It should be noted that the conceptualization should be formed at the right level of abstraction: The model must be sufficiently complex to represent the characteristics or behaviors

---

The journal model is *IEEE Transactions on Biomedical Engineering*.

of the physical system, while at the same time it must be simple enough that the necessary parameters can be well estimated and the optimization problem is computationally tractable [2].

In translational science, most research is carried out on the molecular level which study the fundamental building blocks of living organisms—DNA, RNA and protein as well as their interactions. The terms "genomics" and "proteomics" were coined for the two major branches concerning the study of genomes and proteins of organisms, respectively. Ever since the the advent of DNA microarrays in the mid-1990s, the development of related analytical equipments and methods has boomed. Microarray and next generation sequencing for genome-wide gene profiling, mass spectrometry for large scale protein analysis, along with other high-throughput technologies greatly expand the experimental capabilities and propel the research.

In this chapter, the high-throughput technologies related to the research conducted for the dissertation are briefly reviewed. The challenges in genomic and proteomic data analysis and the problems with existing methods are outlined, which bring forward the proposed methodology for biomarker detection and systematic analysis.

A. MS-based proteomics

Mass spectrometry is a key analytical tool in proteomics. It is widely used for large-scale protein profiling with applications in biomarker discovery [6], signaling pathway monitoring [7, 8], drug development, and disease classification [9]. A mass spectrometer measures the concentration of ionized molecules at a range of mass-to-charge ratios (m/z). MS instruments consist of three modules: an ionization source, a mass analyzer and a detector which captures the ions and measures the intensity of each ion species. Widely used ionization methods include electrospray ionization (ESI) [10] and

matrix-assisted laser desorption/ionization (MALDI) [11, 12]. Mass analyzers separate the ions according to their mass-to-charge ratios. There are several types of mass analyzers including the Orbitrap [13], Quadrupole [14], Time-of-Flight (TOF) [15,16], and Fourier transform ion cyclotron resonance (FTICR) [17].

In a typical MS experiment, unknown protein mixtures extracted from biological samples are first digested into peptides by enzymes such as trypsin. Then peptides enter a mass spectrometer where they get ionized and separated according to their mass-to-charge ratios. As a result, a spectrum is produced which plots the ion intensity against the mass-to-charge ratio. The recorded intensities reflect the abundance or concentration of peptides. Liquid Chromatography (LC) is often coupled with MS to achieve additional separation of peptides and thus reduce the complexity of an individual mass spectrum: before entering the mass spectrometer, peptides are first passed through an LC column where they are separated by the retention time, depending on their physicochemical properties and interactions with the solvent. A single LC-MS experiment usually produces hundreds to thousands of mass spectra sampled during the LC elution process.

Analysis of LC-MS experiments by computational methods is challenging due to the huge data size and rich information content, and moreover is complicated by several facts including: (1) Proteins contained in complex samples such as plasma and tissue extracts have a wide dynamic concentration range (e.g. 10 orders of magnitude), plus peptides differ in ionization efficiencies, which means that the observed peptide signal from MS data may also have a wide dynamic range. While high abundance peptides are relatively easy to be identified, low abundance peptides/proteins, which are often of more biological importance, are likely to be buried under noise or interfering signals and thus hard to be detected [18]. (2) The shape of peptide chromatographic peaks is not well predicted [19]. Due to experiment settings and the

nature of the analytes, asymmetric shape or plateaus of chromatographic peaks may be observed, which requires designed detection algorithms to be robust in tracking signals from various peptide species and be adaptable across experiments. (3) A peptide species may register several groups of peaks in different regions of the spectra due to the following two points: first, a peptide species may take various numbers of charges during ionization, therefore its peaks can be observed at different charge states; second, at a given charge state, several peaks with equal spacing can be observed due to heavy isotopes (e.g. ^{13}C), which are commonly referred to as isotopic peaks or the isotope series. Correctly identifying all the peaks and assigning them to the right peptide is a non-trivial task. (4) The signal density can be very high even in high-resolution LC-MS data and overlapping peptide peaks are commonly observed, the detection of which is very challenging.

1. Feature extraction in MS data analysis

Peptide detection and identification, which extracts features from the raw spectra and converts the raw data into a list of peptides, is usually the first step in MS data processing. It is a critical step that directly affects the accuracy of subsequent analysis, such as protein identification and quantification, data alignment between multiple experiments, biomarker discovery, and sample classification.

Fragmentation spectra produced by tandem mass spectrometry (MS2) are frequently used by popular software such as SEQUEST and Mascot [20] for database searching to give peptide identifications. However, only a small percentage of peptides present in the sample get selected for fragmentation analysis, and of these selected peptides even fewer can be correctly identified by database searching due to spectrum matching ambiguity or co-eluting precursor ions [21]. Furthermore, quantitation of peptide abundance based on MS2 spectral counting is quite rough, and highly vari-

able especially for low abundance peptides [22]. (Though by using well established stable isotope labeling approaches such as tandem mass tags, the relative abundance of analytes in different samples can be accurately determined [23].)

Therefore many algorithms for peptide detection are designed to use MS1 information directly, and thus have the potential to identify more peptides. When mass spectra have low resolution in which isotopic peaks cannot be baseline resolved (i.e. the isotopic peaks convolve together to form isotope envelopes, and only one peak can be observed for one peptide at a given charge state), and when peptides are singly charged as commonly observed in MALDI, to report each detected peak as a peptide feature might be sufficient, as done in [24–27]. But for high resolution spectra, reporting each observed peak as a unique peptide species would give rise to too many false positives. Thus a variety of algorithms for deisotoping and charge states deconvolution have been proposed. Such algorithms can be mainly divided into two categories: one-dimensional (1D) algorithms (e.g. NITPICK [28], PepList [29], Decon2LS [30] and Hardklör [31]), which perform peak picking, deisotoping and charge state assignment on a scan-to-scan basis, and two-dimensional (2D) algorithms (e.g. MZmine [32], SpecArray [18], msInspect [33], SuperHirn [34], VIPER [35], MaxQuant [36], and OpenMS [37]), which capture the 2D nature of LC-MS data and utilize information from both the mass-to-charge and retention-time (RT) dimensions for peptide detection. 2D algorithms appear to be more promising in handling LC-MS data. Regardless of category, most of the aforementioned algorithms are grounded on the idea of greedy template-matching which makes them ineffective to detect overlapping and low abundance peptides. This motivates the development of a global optimization based Bayesian peptide detection algorithm for feature extraction and peptide detection.

2. MS analysis pipeline for biomarker discovery

In clinical applications of mass spectrometry, the number of samples available is usually in the range of tens to a few hundred (small sample size). The samples are analyzed by an MS instrument and transformed into a series of mass spectra containing hundreds of thousands of intensity measurements with signal generated by thousands of proteins/peptides (large feature dimension). This small-sample, high-dimensionality problem requires the experiment and analysis to be carefully designed and validated in order to arrive at statistically meaningful results.

The MS analysis pipeline consists of many steps, including sample preparation, instrument analysis, feature extraction, quantification, statistic analysis and so on. The pipeline can be viewed as a noisy channel, where each processing step introduces some loss or distortion to the underlying signal and the biomarker discovery results are affected by the combined effects of all upstream steps. While individual components of the MS pipeline have been studied at length, little work has been done to integrate the various modules, evaluate them in a systematic way, and focus on the impact of the various steps on the end results of differential analysis and sample classification. In real experiments, it is not easy to decouple the compound parameter effects and determine the marginal influence of various modules on the end results, due to variations and the complicated nature of the workflow. Moreover, owing to contaminants and unknown or incomplete ground-truth, it is hard to meaningfully evaluate and compare results across different experiments. Thus we propose a model-based approach to evaluate the pipeline systematically. It allows us to better understand the characteristics of the MS data, the contributions of individual modules, and the performance of the full pipeline.

B. DNA microarray-based genomics

DNA microarrays are small, solid supports onto which the sequences from large numbers of genes are immobilized at fixed locations, known as probes. Based on probe-target hybridization, the microarrays can be used to measure the expression levels of hundreds and thousands of genes simultaneously. It revolutionizes the way scientists examine gene expression, but also poses many challenges in the analysis of the resulting high-dimension small-sample data sets.

Microarray data frequently contain missing values (MVs) because imperfections in data preparation steps (e.g. poor hybridization, chip contamination by dust and scratches) create erroneous and low-quality values, which are usually discarded and referred to as missing. It is common for gene expression data to contain at least 5% MVs and, in many public accessible data sets, more than 60% of the genes have MVs [38].

There exists many imputation methods for estimating MVs. But only a few studies have examined the impact of MV imputation on high level analysis such as sample clustering and classification. Furthermore, these studies are problematic in key steps such as MV generation and classifier error estimation. To address these problems, a model-based approach is developed to explore the relationship among the data quality, MVs and high level analysis in the microarray data analysis pipeline.

C. Organization of the dissertation

This thesis is organized as described below.

Chapter II proposes a Bayesian approach, BPDA, for peptide detection in MS data, such as MALDI-TOF and LC-MS, with high enough resolution [39]. BPDA is based on a rigorous statistical framework and avoids problems, such as voting and

ad-hoc thresholding, generally encountered in algorithms based on greedy template-matching. It systematically evaluates all possible combinations of possible peptide candidates to interpret a given spectrum, and iteratively finds the best fitting peptide signal in order to minimize the mean squared error of the inferred spectrum to the observed spectrum. In contrast to previous methods, BPDA performs deisotoping and deconvolution of mass spectra simultaneously, which enables better identification of weak peptide signals and produces higher sensitivity results. Unlike template-matching algorithms, BPDA can effectively handle overlapping peptide features. Experimental results indicate that BPDA performs well on both simulated data and real data, for various resolutions and signal to noise ratios, and compares very favorably with commonly used commercial and open-source software.

Chapter III proposes a 2D Bayesian peptide detection algorithm, BPDA2d, which extends the previous work [40]. BPDA2d is specially designed for LC-MS. It models the spectra from both m/z and RT dimensions, thereby better capturing and fitting the properties of LC-MS data. Instead of local template matching, BPDA2d performs global optimization for all possible peptide candidates and systematically optimizes their signals. Since BPDA2d looks for the optimal among all possible interpretations of the given spectra, it has the capability in handling complex spectra where features overlap. For each peptide candidate, BPDA2d takes into account its elution profile, charge state distribution, and isotope pattern, and it combines all evidence to infer the signal and existence probability of the candidate. By piecing all evidence together — especially by deriving information across charge states — low abundance peptides can be better identified and peptide detection rates can be improved. Our experiments indicate that BPDA2d outperforms state-of-the-art detection methods on both simulated data and real LC-MS data, according to sensitivity and detection accuracy.

While Chapter II and III focus on enhancing the feature extraction module in the MS analysis pipeline, Chapter IV investigates the entire pipeline from a systems point of view. A model-based approach is presented to integrate various pipeline modules and evaluate the pipeline systematically, by means of simulation with ground-truthed data [41]. Key steps and factors of the pipeline are captured, and their effects on peptide identification rate, protein quantification accuracy, differential expression results, and classification accuracy are studied. The proposed MS-based proteomics framework can be used to optimize the workflow and predict experiment results.

Chapter V extends the system approach presented in Chapter IV to the analysis of DNA microarray data. In this chapter, a model-based approach is developed to examine the effects of MVs and their imputation on classification in a complicated microarray data analysis pipeline [42]. Six popular imputation algorithms, two feature selection methods, and three classification rules are considered. The situations when it is suitable to apply MV imputation are identified and recommendations regarding imputation are provided.

Chapter VI summarizes the dissertation and proposes future research directions.

D. Main contributions

The main contributions of this work are summarized below:

- Developed Bayesian peptide detection algorithms to optimize the feature extraction step in MS-based proteomics. The algorithms can effectively identify low abundance peptides and overlapping peptides, which is not satisfactorily addressed by existing approaches. The proposed methods achieved better sensitivity and accuracy results compared to many popular software packages.

- Designed a simulation framework for MS-based proteomics, which enables sys-

tematic evaluation of the MS data analysis pipeline. By contrast, in previous methods the pipeline is frequently chopped up into individual modules, and is rarely studied and assessed as a whole from a systems point of view. The proposed framework can be used to determine the working range of important parameters, aid experimental design, predict the biomarker discovery results, and to pinpoint critical bottlenecks which are worth investing resources into for improving performance.

- Proposed a model-based approach to examine how different properties of a microarray data set influence the quality of the imputed data and how missing value imputation influence the classification performance. The results suggest that it is beneficial to apply MV imputation when the noise level is high, variance is small, or gene-cluster correlation is strong, under small to moderate MV rates. In addition, an MV-rate related peaking phenomenon was uncovered.

CHAPTER II

BAYESIAN PEPTIDE DETECTION FOR MASS SPECTROMETRY*

A. Background

Feature extraction, which includes peptide detection and quantification, is usually the first step in MS data processing pipeline. Accurate detection and quantification of peptides and proteins is essential for biomarker discovery, drug development and disease classification.

A variety of algorithms for peptide detection in high resolution mass spectra have been proposed. Most of these algorithms are grounded on the idea of greedy template-matching. Such algorithms include PepList [29], Decon2LS [30], Noy's method [43], MZmine [32], SpecArray [18], msInspect [33], SuperHirn [34], VIPER [35], and OpenMS [37]. The templates used are often based on theoretic isotope patterns calculated from peptide masses [44]. If an observed group of peaks matches the proposed template well — the quality of the match is usually assessed by a fitting score — it will be reported as a feature and then subtracted from the spectra. The matching and subtraction process goes on until no more matches can be found. The major problem with greedy template-matching is that it may be ineffective to detect overlapping peptides. In the case of overlapping (e.g. one doubly charged peptide can overlap with a singly charged peptide of half the mass given that the two elute from chromatography column at a similar time), if the peak group of one peptide is incorrectly matched and subtracted, the rest of the overlapping peptides cannot be

---

*Reprinted with permission from "BPDA—a Bayesian peptide detection algorithm for mass spectrometry" by Y. Sun, J. Zhang, U. M. Braga-Neto, and E. R. Dougherty, *BMC Bioinformatics*, vol.11, pp. 490-500, 2010, Copyright 2010 by BioMed Central.

detected correctly using the remaining signal, which may result in error propagation. Besides, each template is aimed at matching isotopic peaks of one single peptide, and thus are likely to be different from the observed overlapping peaks, which renders a poor match and reduces the sensitivities of these algorithms. Alternatives to greedy template-matching based approaches include 1D algorithms such as NITPICK, which is based on non-greedy regression; Hardklör, which approximates an isotope peak cluster by a set of averagine models [45]. They also include 2D algorithms such as MaxQuant, which mainly relies on the distance among isotope peaks and the correlation between isotope labeled (SILAC) pairs to detect and quantify peptides in SILAC-proteome experiments.

In this chapter, we propose a Bayesian Peptide Detection Algorithm (BPDA) to optimize the workflow of peptide deisotoping and charge state deconvolution. BPDA can be applied to data generated by MS instruments with mass resolutions high enough to baseline-resolve isotopic peaks. It evaluates all possible combinations of possible peptide candidates (originated from well-defined peaks of the raw spectrum — see Methods section for more details) to interpret a given spectrum, and iteratively finds the best fitting peptide parameters (peptide peak heights, existence probabilities, etc.) in order to minimize the mean squared error (MSE) of the inferred spectrum to the observed spectrum.

B. Methods

For 1D MS spectrum, we first perform spectrum preprocessing to remove the baseline, filter the noise and generate a list of peptide candidates. Then BPDA is applied based on the developed MS model to infer the best fitting peptide signals of the observed spectrum, the results being peptide abundances, existence probabilities and so on.

For 2D LC-MS spectra, we first detect peptide elution peaks along the retention time dimension, and build elution peak groups by collecting the peaks which have similar retention time together using a method similar to [46]. Each group contains a series of consecutive spectra, which are then averaged to form a mean spectrum. The rationale of using a mean spectrum to represent the group is that the noise of consecutive spectra could be canceled out to a certain degree [24]. The BPDA algorithm is then applied to each of the mean spectra, and finally an overall peptide list is generated. The details of the preprocessing step, the proposed MS model, and the BPDA algorithm are described in the following subsections.

1. Spectrum preprocessing and obtaining peptide candidates

A non-flat baseline is often observed in mass spectra, the presence of which can distort the true signal pattern. Thus the first preprocessing step is to detect and subtract the baseline from MS spectra. We use the minimum of a sliding window along the m/z axis as the baseline, similar to the method used in [47]. The next step is peak detection. We use the Matlab function "mspeaks" (`http://www.mathworks.com/access/helpdesk/help/toolbox/bioinfo/ref/mspeaks.html`) to perform this task. The algorithm first identifies all local maxima in the wavelet denoised spectrum as putative peak locations. Then peaks are filtered based on their intensities and signal to noise ratios. The last step of preprocessing is to obtain a list of peptide candidates. Considering one detected peak with centroid at m/z value d, we want to find out which peptides can potentially register a peak at this position. The answer is given below in terms of the masses of such peptides:

$$mass = i(d - m_{pc}) - j\, m_{nt},\ i = 1, 2, \ldots, cs,\ j = 0, 1, \ldots, iso, \tag{2.1}$$

where $mass$ is the mass of one peptide candidate, m_{pc} is the mass of one positive charge and m_{nt} is the mass shift caused by addition of one neutron. Due to mass defect, the mass shift varies for different elements. We approximate m_{nt} using the mass shift from ^{13}C to ^{12}C, which is 1.0034, since Carbon contributes most to the isotope patterns. This approximation works well if the mass calibration of the instrument is correct. The parameters cs and iso are user defined maximum numbers of considered charge states and isotopic positions, respectively. It is easy to see from the above equation that each detected peak gives rise to $cs \times (iso + 1)$ different peptide candidates (masses). These candidates exhaust all the possibilities to generate the peak with centroid d, but it does not follow that all the candidates really exist in the sample. Therefore, our primary goal in peptide detection is to find the existence probability of each peptide candidate. Also note that the total number of candidates should be less than or equal to $cs \times (iso+1) \times$ number of detected peaks, as is possible that multiple peaks yield the same candidate mass.

2. Modeling the mass spectrum

Suppose N peptide candidates are obtained from the observed spectrum using the method described in the previous section. Each candidate can generate a series of peaks over different charge states, and at each charge state several isotopic peaks can be registered. The signal generated by the kth peptide candidate is thus modeled by the following equation, in which i and j represent the charge state and the isotopic position of the candidate peptide, respectively:

$$g_k(x_m) = \sum_{i=1}^{cs} \sum_{j=0}^{iso} c_{k,ij} f(x_m; \rho_{k,ij}, \alpha_{k,ij}),\ m = 1, 2, \ldots, M, \tag{2.2}$$

where the peak shape function is given by $f(x_m; \rho_{k,ij}, \alpha_{k,ij}) = e^{-\rho_{k,ij}(x_m - \alpha_{k,ij})^2}$.

That is, the peak is modeled as Gaussian-shaped, as in [19]. It is reported that the Gaussian-shaped peak approximates the reality well enough to obtain good detection results [43]. Still, this peak shape function can be adjusted for different instruments without affecting the overall structure of the algorithm.

The observed spectrum is a mixture of the signal generated by the N peptide candidates plus Gaussian random noise, which can be modeled as:

$$y_m = \sum_{k=1}^{N} \lambda_k g_k(x_m) + \epsilon_m = \sum_{k=1}^{N} \lambda_k \sum_{i=1}^{cs} \sum_{j=0}^{iso} c_{k,ij} f(x_m; \rho_{k,ij}, \alpha_{k,ij}) + \epsilon_m, \qquad (2.3)$$

$$m = 1, 2, \ldots, M.$$

In the above three equations, x_m is the mth mass-to-charge ratio (m/z) in the spectrum, y_m is the observed intensity at x_m, M is the number of observations, and ϵ_m is Gaussian random noise with zero mean and standard deviation σ. The value of σ can be approximated by the standard deviation of the background region in the spectrum. Note that we model ϵ_m as additive Gaussian which is generally a good model for the thermal noise in electronic instruments. There are reports of non-Gaussian noise in FTMS [48] and thus it is safer to apply the proposed algorithm to TOF MS instruments [49]. The parameters of the kth candidate, namely, $\alpha_{k,ij}$, $\rho_{k,ij}$, λ_k and $c_{k,ij}$ are discussed in detail below:

- $\alpha_{k,ij}$ is the theoretic centroid (m/z value) of the peak generated by candidate k, at charge state i and isotopic number j.

  $$\alpha_{k,ij} = \frac{mass_k + i\, m_{pc} + j\, m_{nt}}{i}, \quad i = 1, 2, \ldots, cs,\ j = 0, 1, \ldots, iso, \qquad (2.4)$$

  where $mass_k$ is the mass of the kth candidate. Since the candidate's mass is already obtained, $\alpha_{k,ij}$ can be calculated.

- $\rho_{k,ij}$ relates to the shape (width) of the peak centered at $\alpha_{k,ij}$. It can be estimated by using its relationship to the peak's Full Width at Half Maximum (FWHM): $\rho_{k,ij} = 2\sqrt{2\ln 2}/\text{FWHM}$.

- λ_k is an indicator random variable, which is 1 if the kth peptide candidate truly exists in the sample and 0 otherwise.

- $c_{k,ij}$ is the height (i.e. intensity) of the peak generated by peptide k, at charge state i and isotopic number j.

In summary, the model considers peaks at different isotopic positions and charge states simultaneously for each peptide candidate, incorporating candidates' existence probabilities and the spectrum thermal noise.

3. Bayesian peptide detection

Let

$$\boldsymbol{\theta} \triangleq \{\lambda_k, c_{k,ij}; k = 1, \ldots, N,\ i = 1, \ldots, cs,\ j = 0, \ldots, iso\}$$

be the set of all the unknown model parameters. The goal of our algorithm is to determine the value of $\boldsymbol{\theta}$ based on the observed spectrum $\mathbf{y} = [y_1, \ldots, y_M]^T$. In fact, the value of λ_k is of our prime interest for the peptide detection problem. For this purpose, we can use a Bayesian approach to first obtain the *a posteriori* probability (APP) of all the parameters, $P(\boldsymbol{\theta}|\mathbf{y})$. Then the APPs $P(\lambda_k|\mathbf{y}), k = 1, \ldots, N$, can be obtained by integration of the joint posterior distribution $P(\boldsymbol{\theta}|\mathbf{y})$ over all parameters except λ_k. Clearly, the calculation involves high dimension integration which is not an easy task. Besides, due to the highly nonlinear nature of the data model, none of the desired APPs can be obtained analytically. To overcome the computational obstacle, we resort to the Gibbs sampling method [50], which is a variant of the Markov Chain

Monte Carlo (MCMC) approach [51], to sample the model parameters.

Gibbs sampling is an iterative scheme, which uses the popular strategy of divide-and-conquer to sample a subset of parameters at a time while fixing the rest at the sample values from the previous iteration, as if they were true. In other words, for the lth parameter group $\boldsymbol{\theta}_l$, we sample from the conditional posterior distribution $P(\boldsymbol{\theta}_l|\boldsymbol{\theta}_{-l}, \mathbf{y})$, where $\boldsymbol{\theta}_{-l} \triangleq \boldsymbol{\theta} \setminus \boldsymbol{\theta}_l$. After this sampling process iterates among the parameter groups for a sufficient number of cycles (which is referred to as the "burn-in" period), convergence is reached. The samples collected afterwards are shown to be from the marginal posterior distribution $P(\boldsymbol{\theta}_l|\mathbf{y})$, which is independent of $\boldsymbol{\theta}_{-l}$, and thus these samples can be used to estimate the target parameters.

The Gibbs sampling process for the kth peptide candidate and the derivations of the conditional posterior distributions of important model parameters are detailed below.

a. Sampling the peak height vector

The heights of all the possible peaks (over different charge states and isotopic positions) of the kth peptide candidate are included in the peak height vector $\mathbf{c}_k$, which is defined as $\mathbf{c}_k \triangleq [c_{k,ij}; i = 1, \ldots, cs,\ j = 0, \ldots, iso]^T$. By the Bayesian principle, the conditional posterior distribution of $\mathbf{c}_k$ is proportional to the likelihood times the prior:

$$P(\mathbf{c}_k\,|\mathbf{y}, \boldsymbol{\theta}_{-\mathbf{c}_k}) \propto P(\mathbf{y}|\boldsymbol{\theta})\mathrm{Prior}(\mathbf{c}_k), \tag{2.5}$$

where $\boldsymbol{\theta}_{-\mathbf{c}_k} \triangleq \boldsymbol{\theta} \setminus \mathbf{c}_k$.

It is easy to show the likelihood satisfies

$$P(\mathbf{y}|\boldsymbol{\theta}) \propto \exp\{-\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{G}\lambda^{(0)} - \lambda_k\mathbf{H}_k\mathbf{c}_k)^T\mathbf{I}_{M\times M}(\mathbf{y} - \mathbf{G}\lambda^{(0)} - \lambda_k\mathbf{H}_k\mathbf{c}_k)\}, \tag{2.6}$$

where

$$\boldsymbol{\lambda}^{(s)} \triangleq [\lambda_1, \ldots, \lambda_k = s, \ldots, \lambda_N]^T,\ s \in \{0, 1\}, \tag{2.7}$$

$$\mathbf{G} = \begin{pmatrix} g_1(x_1) & g_2(x_1) & \ldots & g_N(x_1) \\ g_1(x_2) & g_2(x_2) & \ldots & g_N(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ g_1(x_M) & g_2(x_M) & \ldots & g_N(x_M) \end{pmatrix}_{M \times N}, \tag{2.8}$$

with the (m, k)-th entry $g_k(x_m) = \sum_{i=1}^{cs} \sum_{j=0}^{iso} c_{k,ij} f(x_m; \rho_{k,ij}, \alpha_{k,ij})$ representing the signal at m/z value x_m generated by peptide candidate k. In addition,

$\mathbf{H}_k = [h_{m,(i-1)\times(iso+1)+j+1}]_{M \times cs(iso+1)}$, with $h_{m,(i-1)\times(iso+1)+j+1} = f(x_m; \rho_{k,ij}, \alpha_{k,ij}) = e^{-\rho_{k,ij}(x_m - \alpha_{k,ij})^2}$.

The heights of the isotopic peaks of peptide candidate k at charge state i follow a multinomial distribution [52], which by the Central Limit Theorem can be approximated by a Gaussian distribution as below:

$$\begin{aligned} P(c_{k,ij}, j = 0, \ldots, iso \,|a_k, \eta_{k,i}, \boldsymbol{\pi}_k) &= MN(a_k \eta_{k,i}, \boldsymbol{\pi}_k) && (2.9) \\ &\approx N(a_k \eta_{k,i} \boldsymbol{\pi}_k, \\ &\qquad a_k \eta_{k,i} [\mathrm{diag}(\boldsymbol{\pi}_k) - \boldsymbol{\pi}_k^T \boldsymbol{\pi}_k]), && (2.10) \end{aligned}$$

where a_k is the total centroid intensity of candidate k, and $\boldsymbol{\eta}_k \triangleq [\eta_{k,1}, \eta_{k,2}, \ldots, \eta_{k,cs}]^T$ and $\boldsymbol{\pi}_k \triangleq [\pi_{k,0}, \pi_{k,1}, \ldots, \pi_{k,iso}]^T$ denote the charge state distribution and the theoretical isotopic distribution of peptide candidate k, respectively.

Thus the prior distribution of the peak height vector $\mathbf{c}_k$ is given by:

$$\mathrm{Prior}(\mathbf{c}_k) = P(\mathbf{c}_k \,|a_k, \boldsymbol{\eta}_k, \boldsymbol{\pi}_k) \approx N(\boldsymbol{\mu}_{\mathbf{c}_k}, \boldsymbol{\Sigma}_{\mathbf{c}_k}), \tag{2.11}$$

where

$$\boldsymbol{\mu}_{\mathbf{c}_k} = [a_k\eta_{k,1}\boldsymbol{\pi}_k^T, a_k\eta_{k,2}\boldsymbol{\pi}_k^T, \ldots, a_k\eta_{k,cs}\boldsymbol{\pi}_k^T]^T, \tag{2.12}$$

$$\boldsymbol{\Sigma}_{\mathbf{c}_k} = \text{diag}(\Sigma_i), \tag{2.13}$$

with

$$\Sigma_i = a_k\eta_{k,i}[\text{diag}(\boldsymbol{\pi}_k) - \boldsymbol{\pi}_k^T\boldsymbol{\pi}_k],\ i = 1, 2, \ldots, cs. \tag{2.14}$$

Substituting Eq. 2.6 and Eq. 2.11 into Eq. 2.5 and it can be shown by algebraic manipulations [53] that the conditional posterior distribution of $\mathbf{c}_k$ is also Gaussian, with the mean vector and covariance matrix given below:

$$\boldsymbol{\Sigma}_{\mathbf{c}_k|\mathbf{y},\boldsymbol{\theta}_{-\mathbf{c}_k}} = (\mathbf{I} - \mathbf{K}\mathbf{H}_k)\boldsymbol{\Sigma}_{\mathbf{c}_k}, \tag{2.15}$$

$$\boldsymbol{\mu}_{\mathbf{c}_k|\mathbf{y},\boldsymbol{\theta}_{-\mathbf{c}_k}} = \boldsymbol{\mu}_{\mathbf{c}_k} + \mathbf{K}(\mathbf{y} - \mathbf{G}\lambda^{(0)} - \mathbf{H}_k\boldsymbol{\mu}_{\mathbf{c}_k}), \tag{2.16}$$

where $\mathbf{K} \triangleq \boldsymbol{\Sigma}_{\mathbf{c}_k}\mathbf{H}_k^T\left(\mathbf{H}_k\boldsymbol{\Sigma}_{\mathbf{c}_k}\mathbf{H}_k^T + \sigma^2\mathbf{I}_{M\times M}\right)^{-1}$ is known as the Kalman gain matrix [54].

b. Sampling the total centroid intensity

The total centroid intensity of candidate k is denoted by a_k, whose conditional distribution takes different forms for different values of λ_k.

When $\lambda_k = 1$ (the kth candidate is inferred to be present), by definition,

$$a_k\,|(c_{k,ij}, \lambda_k = 1) = \sum_{i=1}^{cs}\sum_{j=0}^{iso} c_{k,ij} \cdot I_{c_{k,ij}>0}. \tag{2.17}$$

When $\lambda_k = 0$ (the kth candidate is inferred to be absent), the distribution of a_k, which is independent of the observation $\mathbf{c}_k$, is modeled by a uniform distribution as

below:

$$P(a_k \,|c_{k,ij}, \lambda_k = 0\,) = \text{Unif}(0, u_k), \tag{2.18}$$

where u_k is the upper bound of a_k.

c. Sampling the charge state distribution

Let $\boldsymbol{\eta}_k \triangleq [\eta_{k,1}, \eta_{k,2}, \ldots, \eta_{k,cs}]^T$ denote the charge state distribution of peptide candidate k. Unlike the isotopic distribution, the charge state distribution cannot be theoretically predicted even when the peptide sequence is given. Thus $\boldsymbol{\eta}_k$ needs to be estimated by the Gibbs sampling process. Let $\boldsymbol{b}_k \triangleq [b_{k,1}, b_{k,2}, \ldots, b_{k,cs}]^T$, where $b_{k,i}$ is the total centroid abundance of peptide k at charge state i. Given the charge state distribution and the total centroid abundance of peptide k, the likelihood of $\boldsymbol{b}_k$ is multinomial:

$$P(\boldsymbol{b}_k|\boldsymbol{\eta}_k, a_k) = \text{MN}(a_k, \boldsymbol{\eta}_k). \tag{2.19}$$

As is well known, the conjugate prior to a multinomial likelihood is Dirichlet, which is also a reasonable choice for the prior of $\boldsymbol{\eta}_k$. Thus, let the prior of $\boldsymbol{\eta}_k$ be a Dirichlet distribution with parameter $w\boldsymbol{\alpha}$, where w is a weight parameter that controls the strength of the prior information. A small w is preferable if uncertainty resides in the prior, and vice versa. Then the posterior distribution of $\boldsymbol{\eta}_k$ is given by

$$\begin{aligned} P(\boldsymbol{\eta}_k \,|\mathbf{b}_k\,) &\propto P(\mathbf{b}_k \,|\boldsymbol{\eta}_k)\text{Prior}(\boldsymbol{\eta}_k) && (2.20)\\ &= \text{Dirichlet}(w\boldsymbol{\alpha} + \mathbf{b}_k). && (2.21) \end{aligned}$$

d. Sampling the peptide existence indicator variable

The conditional posterior distribution of λ_k, the existence indicator variable of peptide k, is given by

$$\begin{aligned} P(\lambda_k \,|\mathbf{y}, \boldsymbol{\theta}_{-\lambda_k}) &\propto P(\mathbf{y}\,|\theta)\text{Prior}(\lambda_k) \\ &\propto \exp\{-\frac{1}{2\sigma^2}\|\mathbf{y}-\mathbf{G}\boldsymbol{\lambda}\|^2\}\text{Prior}(\lambda_k), \end{aligned} \tag{2.22}$$

where $\mathbf{G}$ is defined in Eq. 2.8.

The log-likelihood ratio (LLR) of λ_k can be calculated as below

$$\begin{aligned} LLR_{\lambda_k} &= \ln \frac{P(\lambda_k = 1\,|\mathbf{y}, \boldsymbol{\theta}_{-\lambda_k})}{P(\lambda_k = 0\,|\mathbf{y}, \boldsymbol{\theta}_{-\lambda_k})} \\ &= -\frac{1}{2\sigma^2}(\|\mathbf{y}-\mathbf{G}\boldsymbol{\lambda}^{(1)}\|^2 - \|\mathbf{y}-\mathbf{G}\boldsymbol{\lambda}^{(0)}\|^2) + \ln\frac{P(\lambda_k=1)}{P(\lambda_k=0)}, \end{aligned} \tag{2.23}$$

where $\boldsymbol{\lambda}^{(s)}, s \in \{0,1\}$ is defined by Eq. 2.7.

If no prior knowledge is available about which peptide candidates are more likely to be present in the sample, then a reasonable choice for the prior of λ_k could be the uniform distribution. Therefore the last term in Eq. 2.23 can be dropped. The conditional posterior distribution of λ_k is then obtained based on the log-likelihood ratio as follows:

$$P(\lambda_k = 1\,|\mathbf{y}, \boldsymbol{\theta}_{-\lambda_k}) = \frac{1}{1+e^{-LLR_{\lambda_k}}}, \tag{2.24}$$

$$P(\lambda_k = 0\,|\mathbf{y}, \boldsymbol{\theta}_{-\lambda_k}) = 1 - P(\lambda_k = 1\,|\mathbf{y}, \boldsymbol{\theta}_{-\lambda_k}). \tag{2.25}$$

The complexity of the proposed Gibbs sampling algorithm is determined by two factors: (1) the sheer number of peptide candidates, and (2) the correlation between parameters that need to be sampled. The algorithm complexity grows exponentially with the number of peptide candidates, and the correlation between parameters reduces the sampling efficiency. To address these two issues, we first partition non-

overlapping peptide candidates into different groups. The proposed algorithm can be applied to each group in a parallel manner and the algorithm complexity is reduced, because within each group the number of candidates is reduced, and the corresponding signal-containing spectrum region is restricted. Peptide candidates within each group are then clustered by the k-means clustering algorithm [55], the distance measure being the correlation between peptide candidate signals. Peptide candidates within a cluster have strong correlations among each other, and their indicator variables are sampled from the joint conditional posterior distribution. These two measures improve the overall efficiency of the algorithm.

Samples taken after convergence can be used to estimate the target parameters. Particularly, the existence probability of peptide k is calculated as

$$P(\lambda_k = 1|\mathbf{y}) = \frac{1}{R - r_0 + 1} \sum_{r=r_0}^{R} \lambda_k^r \,, \tag{2.26}$$

where r_0 is the first iteration after convergence is reached, R is the total number of iterations, and λ_k^r is the sample value of λ_k in the rth iteration. The kth peptide candidate is said to be detected if its existence probability $P(\lambda_k = 1|\mathbf{y})$ is greater than a predefined threshold.

C. Results

We report below the observed performance of BPDA, side by side with well-known tools, such as OpenMS and Decon2LS, in a number of experiments using both synthetic and real data.

1. Synthetic data

It is difficult to evaluate the performance of a given detection method using real data due to the existence of unpredictable contaminants and the unknown true composition of the samples. The merit of using simulated data is that the ground truth is known and thus algorithm evaluation can be carried out [19, 49].

a. Synthetic 20-mix spectra with different abundance levels (SNRs)

First, to test the robustness of our algorithm, we generated MS data sets with different signal to noise ratios (SNRs), using the method described in [19]. In fact, the mean signal strength (i.e., peptide abundance) was varied while the noise level (i.e., the mean and variance of the noise) was fixed. For each peptide abundance level $a, a \in \{500, 2500, 12500\}$, the simulation was repeated 50 times. In each repetition, 20 true peptides (with abundance level a and masses randomly selected from a quality-control Shewanella Oneidensis data set provided by PNNL (`http://omics.pnl.gov`) served as the input of the data model given by Eq. 2.3. The charge state distribution of one peptide was modeled by a binomial distribution, which was reported to approximate the real data well [19]. The isotopic distribution was obtained for each peptide by using the Averagine model [56] and the Mercury algorithm [44]. The output consists of a simulated mass spectrum. BPDA was applied to obtain the peptide existence probabilities and abundance results. Its performance was evaluated by the classic Receiver Operating Characteristic (ROC) curve. To obtain the ROC curve, first a series of detection levels τ ranging from 0 to 1 with 0.001 increments was selected. Peptides with existence probabilities not less than τ were said to be detected at this specific detection level. The True Positive Rate (TPR) and False Positive Rate (TPR) were then calculated at each detection level as follows: $TPR = \frac{TruePositive}{TruePositive+FalseNegative}$

and $FPR = \frac{FalsePositive}{FalsePositive+TrueNegative}$. One ROC curve (each point on the curve was a pair of TPR and FPR at one detection level) was plotted for each repetition. And the averaged ROC curve for one abundance level was obtained by averaging all the ROC curves corresponding to the same abundance level. We also applied OpenMS on the same data sets — to do so, we first wrote the simulated MS data into a text file with three columns specified by elution time, m/z, and intensity, respectively. Next, the text file was converted to mzXML (which is a valid input file format for OpenMS) by the FileConverter tool integrated in the OpenMS software package (`http://open-ms.sourceforge.net`). Finally, OpenMS was applied on the mzXML file to give the detection results including detected features and their qualities. The ROC results given by the two algorithms for different abundance levels are shown in Fig. 1.

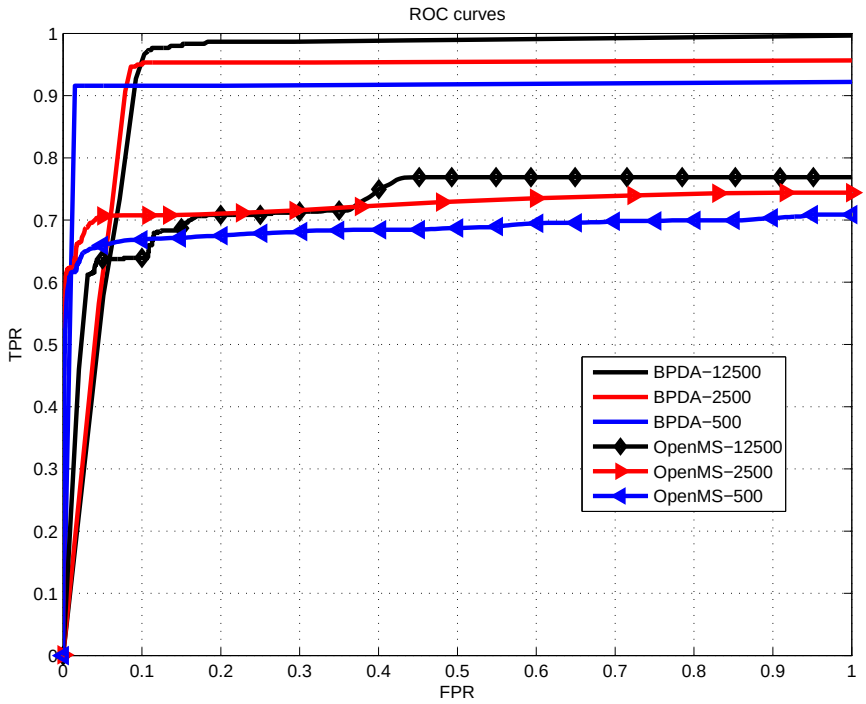


Fig. 1. ROC results for synthetic 20-mix spectra with different abundance levels $a = 500, 2500$ and 12500.

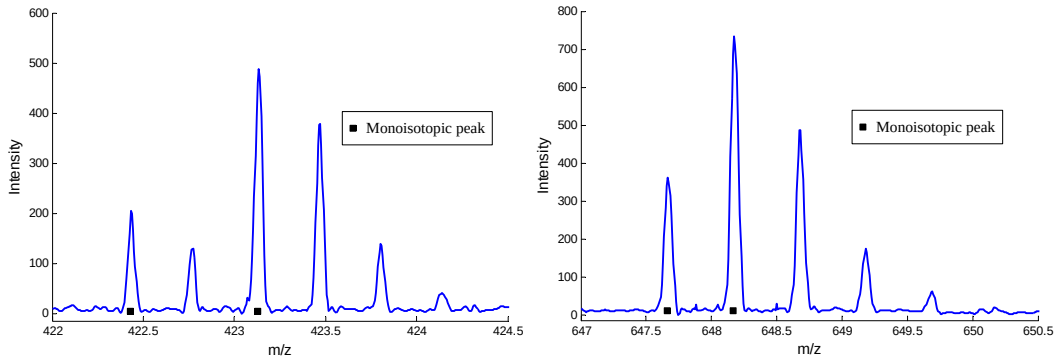
b. Synthetic 10-mix spectrum with overlapping peptides

As noted before, overlapping peptide peaks can complicate the mass spectra and make the detection problem much harder. Thus, we investigated the performance of BPDA in the presence of overlapping peptides. A simulated 10-mix spectrum was generated by 5 pairs of overlapping peptides with unique masses: 1264.279, 1266.383, 1382.247, 1388.367, 1293.323, 1294.345, 1312.441, 1313.451, 1327.386 and 1329.378 Da. The detection results for the comparison between BPDA and OpenMS are summarized in Table I. BPDA detected all 10 peptides when $FPR = 0.1$, with very small mass deviations and quite accurate abundance results. Almost all charge states of the 10 true peptides were correctly reported, except for the highest charge state of the 5th and the 9th peptides. These two charge states were missed because the corresponding peptide signal was very weak. In contrast, when $FPR = 0.1$, OpenMS only detected the 3rd, the 7th and the 9th peptides. And when FPR increased to 0.3, OpenMS achieved its highest TPR (0.6). But it could detect only one pair of peptides (the one with the least overlap) and missed one peptide in each of the other 4 pairs. Two examples are given in Fig. 2 to illustrate the observed overlapping peptide signals and the detection results. The abundance results given by OpenMS were not close to those of the true peptides (although the total abundance of each overlapping pair was not far away from the corresponding total abundance of the true peptides). In total, 18 out of 36 charge states were correctly detected by OpenMS for the 10 peptides, while BPDA correctly detected 34 out of 36, a much larger number.

We remark that Decon2LS results are missing from both synthetic experiments described previously because the synthetic data could not be loaded, causing the program to crash (the data was contained in a mzXML file converted from a 3-column text file by the OpenMS FileConverter tool, whose format was successfully

Table I. Results for the synthetic 10-mix data set with overlapping peptides. Intn, CS and dM denote the normalized intensity, detectable charge states and the mass deviation from the true mass, respectively. When FPR = 0.1, BPDA was able to detect all 10 true peptides, while OpenMS detected only 3 peptides (marked by *). OpenMS achieved its highest TPR (0.6) when FPR = 0.3.

	BPDA	OpenMS
True Mass (Da) / Intn / CS	dM (Da) / Intn / CS	dM (Da) / Intn / CS
1264.279 / 0.034 / 1-3	-0.0065 / 0.032 / 1-3	NA
1266.383 / 0.103 / 1-3	-0.0025 / 0.110 / 1-3	−0.0025 / 0.156 / 1-3
1382.247 / 0.171 / 1-4	0.0028 / 0.181 / 1-4	0.0031* / 0.228 / 1-3
1388.367 / 0.114 / 1-4	-0.0073 / 0.097 / 1-4	−0.0046 / 0.150 / 1-3
1293.323 / 0.006 / 1-3	-0.0081 / 0.007 / 1-2	NA
1294.345 / 0.008 / 1-3	-0.0124 / 0.008 / 1-3	0.0033 / 0.018 / 1-2
1312.441 / 0.229 / 1-4	0.0018 / 0.247 / 1-4	0.0019* / 0.334 / 1-4
1313.451 / 0.183 / 1-4	-0.0061 / 0.173 / 1-4	NA
1327.386 / 0.080 / 1-4	-0.0035 / 0.067 / 1-3	0.0061* / 0.114 / 1-3
1329.378 / 0.072 / 1-4	-0.0035 / 0.078 / 1-4	NA



(a) (b)

Fig. 2. Illustration of overlapping peptides observed in the synthetic 10-mix spectrum. (a) Overlapping peptide signals observed in m/z range 422-424.5, which is generated by monoisotopic masses 1264.28 and 1266.38 at charge state 3. OpenMS missed the first one while BPDA detected both. (b) Overlapping peptide signals observed in m/z range 647-650.5, which is generated by monoisotopic masses 1293.32 and 1294.35 at charge state 2. OpenMS missed the first one while BPDA detected both.

verified against mzXML version 2.1). We contacted Decon2LS's developers, but did not hear from them in time to have the Decon2LS results included.

2. Real data

In this section we report results from experiments carried out with real MS data. The test data and parameter files used for different software tools were provided on the BPDA project website `http://gsp.tamu.edu/Publications/supplementary/sun10a/bpda`. We stick mainly to the recommended parameter values while only adjusted a few parameters such as mass range and detection level to adapt to each data set.

a. MALDI-TOF MS 7-mix spectrum

We tested BPDA on a MALDI-TOF MS 7-mix spectrum, which contained seven standard peptides with monoisotopic masses 1045.535, 1295.678, 1346.728, 1618.815, 2092.079, 2464.191 and 3146.464 Dalton. The spectrum was collected on a Bruker ultraFlex MALDI TOF in the reflectron mode. As stated before, MALDI mostly generates singly charged ions, so we only considered charge state 1 in the test. Since there were contaminants in the data set, the goal was to check whether a detection algorithm could find all the seven true peptides. The detection results of BPDA, Decon2LS, OpenMS, and the commercial software flexAnalysis developed by Bruker Daltonics (`http://www.bdal.de`) are summarized in Table II. BPDA detected the first six peptides with a mean (absolute) mass deviation 0.018 Da. Decon2LS missed the fifth and the last peptides, and the five detected peptides were of a mean mass deviation 0.013 Da. OpenMS missed the forth and the last peptides, and the five detected peptides were of a mean mass deviation 0.025 Da. The commercial software flexAnalysis missed the fifth and the last peptides, and the five detected peptides were of a mean mass deviation 0.013 Da. It can be seen that for the detected peptides, the four algorithms yielded similar intensity results. Only BPDA and OpenMS were able to detect the fifth peptide which had the lowest abundance among the first six peptides. And all methods failed to report the last peptide. Visual inspection suggested that this peptide generated very weak signal and its abundance was about one third of the fifth peptide.

b. High-resolution LC-MS data set MyoLCMS

The preparation of the MyoLCMS data set is detailed as below: the data set was collected from an overnight tryptic digest of horse myoglobin. Capillary liquid chro-

Table II. Results for the MALDI-TOF 7-mix data set. Intn and dM denote the normalized intensity, and the mass deviation from the true mass, respectively.

True Masses (Da)	BPDA dM (Da) / Intn	OpenMS dM (Da) / Intn	Decon2LS dM (Da) / Intn	Bruker dM (Da) / Intn
1045.535	-0.023 / 0.550	0.019 / 0.655	-0.021 / 0.615	-0.023 / 0.532
1295.678	0.003 / 0.173	0.026 / 0.232	0.002 / 0.168	-0.001 / 0.167
1346.728	0.017 / 0.053	0.040 / 0.070	0.013 / 0.050	0.011 / 0.052
1618.815	0.035 / 0.178	NA	0.024 / 0.137	0.022 / 0.202
2092.079	0.021 / 0.004	0.021 / 0.009	NA	NA
2464.191	-0.012 / 0.042	0.020 / 0.034	-0.007/0.030	-0.009 / 0.047
3146.464	NA	NA	NA	NA

matography mass spectrometry (cLC/MS) was performed with a splitless nanoLC-2D pump (Eksigent), a 50 mm-i.d. column packed with 10 cm of 5 mm-o.d. C18 particles, nanoelectrospray and a high-resolution time-of-flight mass spectrometer (MicrOTOF; Bruker Daltonics). The cLC gradient was 2 to 98% 0.1% formic acid/acetonitrile in 172 seconds at 400 nL/min. Sample was injected at a concentration of 60 fmol/mL with an injection volume of 10 mL (600 fmol injected on-column).

There were 172 spectra with a m/z range 44.9 to 3005. To apply BPDA, we first grouped peptide elution peaks, as described in the Method section. A total of 17 groups were obtained, each containing 10-20 consecutive spectra. A mean spectrum was generated for each group, and BPDA was then applied. The detection results of BPDA, OpenMS, and Decon2LS, which was applied in conjunction with VIPER [35],

Table III. Results for the high-resolution LC-MS data set MyoLCMS

	BPDA	OpenMS	Decon2LS
Number of detected monoisotopic masses (features)	1635	2176	823
Average number of charge states for each monoisotopic mass	2.40	1.28	NA
Protein coverage of the top 5% detected features (%)	76.6	29.2	2.0
Protein coverage of the top 40% detected features (%)	81.8	77.9	40.9
No. of horse myoglobin peptides reported in the top 5% detected features	15	3	1
No. of horse myoglobin peptides reported in the top 40% detected features	16	11	7
Mean mass deviation of horse myoglobin peptides in the top 5% detected features (Da)	0.004	0.019	0.020
Mean mass deviation of horse myoglobin peptides in the top 40% detected features (Da)	0.004	0.014	0.014

are summarized in Table III (we also considered the method implemented in the SpecArray package [29], but found it to be inferior to BPDA, OpenMS, and Decon2LS — the results were then omitted for the sake of conciseness). The number of features with unique monoisotopic masses detected by BPDA, OpenMS, and Decon2LS-Viper were 1635, 2176 and 823, respectively. In fact, it is not very informative to evaluate the performance of a detection algorithm solely based on the number of detected features, because of the presence of contaminants and false positive detections. Therefore, we focus on the top detected features yielded by each detection algorithm. Detected features were ranked by quality in descending order. Different algorithms utilize different quality metrics; for example, Decon2LS and OpenMS provide a quality score which measures how well an observed isotope pattern matches the predicted isotope pattern, while BPDA provides the peptide existence probability (see Eq. 2.26) as the quality measure. For each detection algorithm, for a given percentage of top detected features, we calculated the number of detected horse myoglobin peptides and the protein coverage rate. Note that by in-silico digestion of horse myoglobin, there are 39 tryptic peptides with less than 2 missed cleavage sites (19 of which do not contain any missed cleavage sites). Ideally, we should compare algorithms with known peptide composition in the sample and report protein coverage at different false positive rates. However, due to possible peptide contamination in the sample in any LC/MS experiment, actual peptide species presented in the sample are never known and this prevents us from estimating the false positive rates on the reported peptide list. As a result, the statistical significance of reported peptides by different peptide identification algorithms cannot be evaluated and the only option left for users in hope of obtaining a list of peptides with relatively low false positive rate is by applying a percentage threshold on the quality score reported by different algorithms. Thus, protein coverage v.s. percentage threshold on quality score is a meaningful

measurement of the performance of peak detection algorithms and the results are shown in Fig. 3. We need to point out that although the protein coverage of OpenMS seems to be comparable with the proposed algorithm in regions where the quality score percentage threshold is large, in such regions the reported peptide list may contain a lot of false positives and it is not an indication of good or bad algorithm performance. Instead, how quickly an algorithm reaches high protein coverage as the percentage threshold increases should be the measurement of the performance. In Fig. 3, we can see that BPDA reaches high protein coverage much faster than other algorithms at low percentage threshold regions.

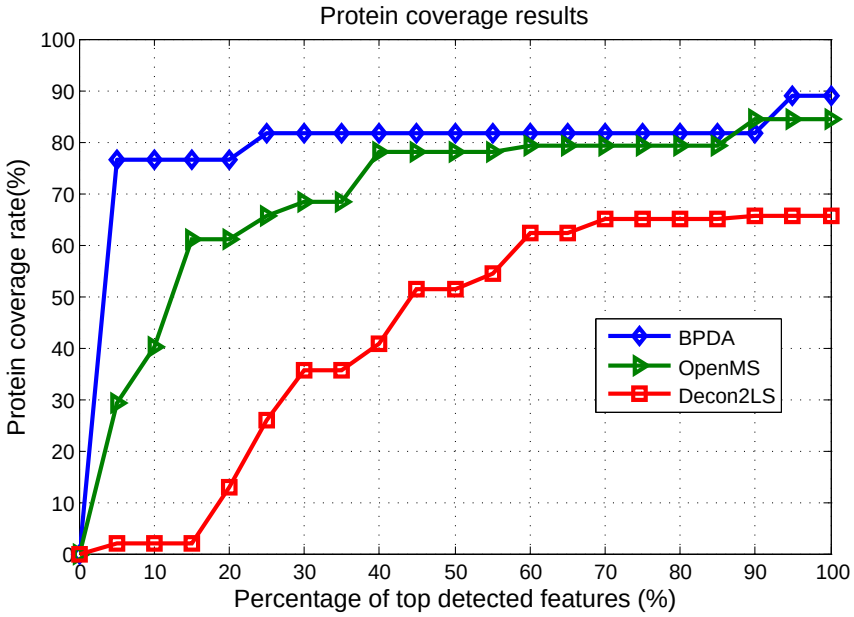


Fig. 3. Protein coverage results achieved by BPDA, OpenMS, and Decon2LS for the LC-MS data set MyoLCMS.

D. Discussion

We observed in our experiments that BPDA performs well on both simulated data and real data, for various SNRs and resolutions, and in complex cases where features overlap.

For the synthetic 20-mix experiment, we observe in Fig. 1 that the sensitivity (i.e., TPR) of BPDA was consistently higher than that of OpenMS for each abundance level, and both methods gave better sensitivity results as the abundance level (i.e., SNR) increased. Also it is observed that BPDA was quite robust for different SNRs. For the synthetic 10-mix experiment with overlapping peptides, we saw that BPDA detected all the peptides at a small false-positive rate $FPR = 0.1$, with very small mass deviations and quite accurate abundance results, and nearly all the charge states of the 10 true peptides were correctly reported. In contrast, at $FPR = 0.1$, OpenMS could detect only a few of the peptides. The abundance results given by OpenMS were not very close to those of the true peptides. Also OpenMS could only detect about half of the charge states.

The results obtained with real data corroborated the findings made with the synthetic experiments. For the MALDI-TOF MS 7-mix data, the four algorithms yielded similar intensity results, but BPDA was the only one to detect six out of the seven peptides. For the MyoLCMS experiment, we focused on protein coverage results, which is an important criterion to determine the confidence in protein identification and quantification [57, 58]. It was observed that BPDA displayed the largest protein coverage among the programs tested.

CHAPTER III

BAYESIAN PEPTIDE DETECTION FOR LC-MS*

A. Introduction

For peptide detection in LC-MS data, 2D algorithms which utilize information from both the mass-to-charge and retention-time dimensions can better capture the features in the data, and thus appear to be more promising compared to 1D algorithms which perform peak picking, deisotoping and charge state assignment on a scan-to-scan basis.

In this chapter, we present BPDA2d, a 2D Bayesian peptide detection algorithm and an extension of BPDA, to process high-resolution LC-MS data more efficiently. BPDA2d shares the core idea with BPDA, which is to systematically evaluate all possible combinations of peptide candidates for spectra interpretation, and to optimize all peptide signals in order to minimize the MSE between inferred and observed spectra. The outputs include peptide monoisotopic mass, retention time, abundance, existence probability, etc. BPDA2d essentially differs from BPDA by explicitly exploiting information residing in the RT dimension to analyze spectra and detect peptides. While BPDA only models peptide signals along the m/z dimension, BPDA2d models the spectra from both m/z and RT dimensions, thereby capturing and fitting the properties of LC-MS data better.

BPDA2d offers following advantages over conventional methods:

(i) BPDA2d carries out global optimization instead of local template matching.

---

*Reprinted with permission from "BPDA2d— a 2D global optimization based Bayesian peptide detection algorithm for LC-MS" by Y. Sun, J. Zhang, U. M. Braga-Neto, and E. R. Dougherty, *Bioinformatics*, vol. 28, pp. 564-572, 2012 Copyright 2012 by Bioinformatics.

It is "global" in two senses: First, for the detection of one peptide candidate, BPDA2d extracts all relevant information and observations (including isotopic peaks, charge state distributions and LC elution peaks) that span all over the m/z-RT space, and pieces all evidence together to infer the candidate's existence probability. As a result, low abundance peptides can be better identified. In contrast, existing algorithms often perform peptide deisotoping at a single charge state, isolating useful information that can be drawn from other charge states. While high abundance charge states may be correctly detected, low abundance charge states might be missed or wrongly assigned. Additional benefits of collating all charge states are discussed in [59] (though their method requires the peak clusters at various charge states to have a moderate correlation, and thus may not work efficiently if the shape of the peak cluster at any charge state differs from other charge states due to the presence of interfering peptides.) Second, BPDA2d performs global optimization for all candidates and simultaneously finds their best fitting signals. Since BPDA2d looks for the optimal among all possible interpretations of the MS spectra, the procedure is thus systematic. In contrast, greedy template-matching based methods detect peptides one by one in a greedy manner, which prevents them from evaluating all potential interpretations of the given spectra and may lead to poor detection of overlapping peptides (See Results section). Therefore, the results are often suboptimal.

(ii) BPDA2d provides existence probabilities for all the candidates considered, as opposed to the fitting scores generally provided by greedy template-matching methods. The metrics used for fitting score calculation may be heuristic (e.g. KL distance [33]). In addition, the range of the fitting score may vary from experiment to experiment, making it hard for the end user to interpret and to select a proper threshold to filter out low quality features. On the contrary, existence probabilities given by BPDA2d are derived based on a solid statistical framework and can be directly

used for probability-based evaluation.

(iii) BPDA2d is flexible in the sense that it makes little assumptions about the underlying spectra. When modeling peptide signals, the model is derived from observations as opposed to employing any pre-assumed peptide peak shape as in [37, 60]. Therefore, BPDA2d is more effective in tracking signals from various peptide species and more adaptable across experiments.

(iv) Most parameters in the proposed method possess a clear physical meaning as they come directly from observations of the mass spectra.

B. Methods

We first preprocess the spectra to remove baseline, filter noise, detect peaks in the m/z-RT plane, and generate a list of peptide candidates annotated by mass and RT. Then BPDA2d is applied based on the developed MS model to infer the best fitting peptide signals of observed spectra, the results being peptide monoisotopic mass, RT, abundance, existence probability, etc. Details of preprocessing steps, proposed MS model, and BPDA2d algorithm are described in the following subsections.

1. Spectra preprocessing and obtaining peptide candidates

Non-flat baselines are often observed in mass spectra. Their presence can distort the true signal pattern. Thus, the baseline of each MS scan is first identified as the running minima along the m/z axis using a window size of 4 Dalton (a tunable parameter), and subtracted from the scan. Then each scan is smoothed by the LOWESS regression method (Matlab mslowess function: `http://www.mathworks.com/help/toolbox/bioinfo/ref/mslowess.html`) with Gaussian kernel and a span of 9 consecutive points.

The next step is 1D peak detection along the m/z axis. We followed the approach implemented in the Matlab mspeaks function. Specifically, in each smoothed MS scan, local maxima are first identified as putative peak locations. Then peaks are filtered based on their intensities and signal to noise ratios (defined as the local maximum divided by the minimum of the two neighboring local minima), and peaks that are too close to each other (might occur due to over segmentation) are joined into a single one. The thresholds used for intensity and over-segmentation filters, τ_{intn} and τ_{seg}, respectively, are automatically determined depending on the characteristic of each input MS scan as below:

$$\begin{aligned}\tau_{intn} &= \text{mean}(intn) + \text{sd}(intn),\\ \tau_{seg} &= \min(\frac{200}{\text{resolution}}, 7 \times \text{lower } 10\% \text{ quantile of the space}\\ &\qquad \text{between neighboring m/z values}).\end{aligned}$$

And the SNR threshold is a tunable parameter with default value 3.

Next, the detected 1D peaks in adjacent spectra are connected along the RT dimension: 1D peaks are first sorted by their centroid m/z positions, and then divided into disjoint subsets, in which the maximal m/z distance between two 1D peaks is less than twice the smallest m/z in the subset times Δm (a user defined mass error in ppm). For each subset, the peaks are then sorted/connected according to their RT positions (if multiple peaks have the same RT, only the one with the largest intensity is retained). Next, the connected 1D peaks are split at RT gaps (a tunable parameter), and the resulting so called elution peaks are smoothed by the LOWESS regression method with a ± 3 scan width. The elution peaks could be multimodal, which may for instance be produced by two different peptides with partially overlapping elution peaks, or by isomers with variant post-translational modifications and thus different retention times. Multimodal elution peaks are split at local minima. A point is

identified as a local minima/maxima if it is preceded by a local maxima/minima and is followed by a value greater/lower by 15% (the threshold is a user tunable parameter which should be comparable to the random intensity fluctuations of the instrument). Consequently all elution peaks are now unimodal, which will be used to propose a list of peptide candidates in the next step. For each elution peak, its centroid position in the m/z axis is estimated as the average of the m/z values of the connected 1D peaks weighted by their intensities. This method enables very accurate mass estimation, as reported by [36].

Now, considering one elution peak with centroid at m/z value d, we want to find out which peptide candidates can potentially produce this signal peak. At least two conditions need to be satisfied. (1) The masses of such peptides should be restricted to the following set:

$$\begin{aligned} \{mass \mid mass = i(d - m_{pc}) - j\, m_{nt}, \\ i = 1, 2, \ldots, cs,\ j = 0, 1, \ldots, iso\}, \end{aligned} \tag{3.1}$$

where $mass$ is the mass of such a candidate, m_{pc} is the mass of one positive charge and m_{nt} is the mass shift caused by addition of one neutron. Due to mass defect, the mass shift varies for different elements. We approximate m_{nt} using the mass shift from ^{13}C to ^{12}C, which is 1.0034, since Carbon contributes most to the isotope patterns. But m_{nt} is a user accessible parameter whose value can be changed as needed. The parameters cs and iso are user defined maximal numbers of considered charge states and isotopic positions, respectively. (2) The shapes of such candidates' elution peaks should resemble the aforementioned elution peak with centroid d (hereafter referred to as the "source" elution peak). But in the presence of scan noise, missing values or overlapping peptide signals, the actual shapes of candidates' elution peaks can be quite different from the observed shape of the source peak. Thus, in order to estimate

candidates' elution peaks more accurately, other elution peaks which can be produced by such candidates need to be taken into account. In more detail, assume the source elution peak has given rise to a candidate with mass value $mass_k$ taken from the set defined in Eq. 3.1. Then, theoretically, this candidate can generate a set of elution peaks with centroids given by

$$\begin{aligned} \alpha_{k,ij} &= \frac{mass_k + i\, m_{pc} + j\, m_{nt}}{i}, \\ i &= 1, 2, \ldots, cs,\ j = 0, 1, \ldots, iso, \end{aligned} \tag{3.2}$$

where $\alpha_{k,ij}$ is the theoretic centroid (m/z value) of the elution peak generated by candidate $mass_k$ at charge state i and isotopic number j. In theory, the set of elution peaks generated by this very candidate should have the same shape (up to a multiplicative constant). Therefore, we search in the previous detected elution peaks for those whose centroids are coincident with the values given by Eq 3.2 (within Δm) and have correlation larger than 0.6 with the source elution peak, since these elution peaks can serve as extra evidence to infer the candidate's real elution peak. Finally the candidate's elution peak is estimated by taking the average of all identified elution peaks weighted by the mean intensity of each elution peak involved in the calculation. The candidate's elution profile is then obtained by normalizing its elution peak by the apex, and the corresponding RT of the apex is taken as the candidate's retention time. It is worth to mention that we do not assume any particular shape for candidates' elution profiles, but instead estimate them from relevant observations. Due to heterogeneity of peptides and fluctuations in liquid chromatography, this approach is more robust in the presence of noise and more adaptable across analysis platforms compared to using any pre-defined model [37, 60].

As can be seen from Eq. 3.1, each detected elution peak gives rise to $cs \times (iso + 1)$ different peptide candidates whose elution profiles have been estimated in the

previous step, but it does not follow that all these candidates really exist in the sample. Therefore, our primary goal in peptide detection is to find the existence probability of each peptide candidate. Also note that the total number of candidates should be less than or equal to $cs \times (iso + 1) \times$ (number of detected elution peaks), as it is possible that multiple elution peaks yield the same candidate. It is worth to mention that the way candidates are generated in BPDA2d is fundamentally different from that in BPDA, as additional information carried by elution peaks is utilized. Candidates are now associated with elution profiles in addition to mass values.

2. Modeling the mass spectra

We propose a complete model to capture the specific properties of peptides and mass spectra over the entire m/z-RT plane.

Suppose N peptide candidates are obtained from the observed spectra using methods described in the previous section. Each candidate can generate a series of elution peaks over different charge states, and at each charge state several isotopic peaks can be registered. Hence, the signal generated by the k-th peptide candidate is modeled by Eq. 3.3, in which i and j represent the charge state and the isotopic position of the candidate, respectively. The baseline removed and smoothed spectra (see the previous section for details) are a mixture of signals generated by N peptide

candidates plus Gaussian random noise, which are modeled by Eq. 3.4:

$$\begin{aligned} g_k(x_m,t) &= \sum_{i=1}^{cs}\sum_{j=0}^{iso} c_{k,ij} l_k(t) I_{x_m=\alpha_{k,ij}}, \qquad (3.3) \\ y(x_m,t) &= \sum_{k=1}^{N} \lambda_k g_k(x_m,t) + \epsilon(t) \\ &= \sum_{k=1}^{N} \lambda_k \sum_{i=1}^{cs}\sum_{j=0}^{iso} c_{k,ij} l_k(t) I_{x_m=\alpha_{k,ij}} + \epsilon(t), \qquad (3.4) \\ m &= 1,2,\ldots,M,\ t=1,2,\ldots,T. \end{aligned}$$

In the above two equations, x_m is the m-th mass-to-charge ratio in the signal region, i.e., $x_m \in \{m/z$ values of detected elution peaks$\} \bigcup \{m/z$ values of all $\}$. $\{$.candidates' theoretic peaks $\}$, t indexes spectra, M and T are the total number of m/z values and spectra, respectively, $y(x_m,t)$ represents the intensity at point (x_m,t), I is an indicator function, $I_A = 1$ if $A \neq \emptyset$, $I_A = 0$ otherwise, and the noise term $\epsilon(t)$ follows a Gaussian distribution with zero mean and standard deviation $\sigma(t)$, which is generally a good model for thermal noise in electrical instruments. The value of $\sigma(t)$ can be approximated by the standard deviation of the background region in the t-th scan. The parameters of the kth candidate, namely, $\alpha_{k,ij}$, $l_k(t)$, λ_k and $c_{k,ij}$ are discussed in detail below:

- $\alpha_{k,ij}$ is the theoretic centroid position (in the m/z axis) of the elution peak generated by candidate k, at charge state i and isotopic number j, the value of which is given by Eq. 3.2.

- $l_k(t)$ is the normalized elution profile of the k-th peptide candidate, which is already obtained in previous section.

- λ_k is an indicator random variable, which is 1 if the kth peptide candidate truly exists in the sample and 0 otherwise.

- $c_{k,ij}$ is the apex intensity of the elution peak generated by peptide k, at charge state i and isotopic number j.

In summary, the model considers peptides' elution peaks at different isotopic positions and charge states simultaneously, incorporating candidates' existence probabilities and spectra thermal noise.

3. Bayesian peptide detection

Let $\theta \triangleq \{\lambda_k, c_{k,ij}; k = 1, \ldots, N,\ i = 1, \ldots, cs,\ j = 0, \ldots, iso\}$ be the set of all unknown model parameters. The goal of our algorithm is to determine the value of θ based on the observed spectra vector $\mathbf{y} = [y(x_m, t); m = 1, 2, \ldots, M, t = 1, 2, \ldots, T]^T$. It is not an easy task since there are large number of parameters that need to be simultaneously optimized. To overcome the computational obstacle, we resort to the Gibbs sampling method [50] to sample model parameters. The Gibbs sampling process for the kth peptide candidate and the derivations of the conditional posterior distributions of important model parameters are detailed below.

a. Sampling the apex vector

The apex vector $\mathbf{c}_k \triangleq [c_{k,ij}; i = 1, \ldots, cs,\ j = 0, \ldots, iso]^T$ incorporates all possible elution peaks (over different charge states and isotopic positions) of the kth peptide candidate. By the Bayesian principle, the conditional posterior distribution of $\mathbf{c}_k$ is proportional to the likelihood times the prior:

$$P(\mathbf{c}_k \,|\mathbf{y}, \theta_{-\mathbf{c}_k}) \propto P(\mathbf{y}|\theta)\text{Prior}(\mathbf{c}_k), \tag{3.5}$$

where $\theta_{-\mathbf{c}_k} \triangleq \theta \setminus \mathbf{c}_k$.

It is easy to show the likelihood satisfies

$$P(\mathbf{y}|\theta) \propto \exp\{-\frac{1}{2}(\mathbf{y} - \mathbf{G}\lambda^{(0)} - \lambda_k \mathbf{g}_k)^T {\mathbf{\Sigma_e}}^{-1}(\mathbf{y} - \mathbf{G}\lambda^{(0)} - \lambda_k \mathbf{g}_k)\}, \tag{3.6}$$

where

$$\begin{aligned}\mathbf{y} = [y(x_1,1), y(x_1,2), \ldots, y(x_1,T), y(x_2,1), y(x_2,2), \ldots, y(x_2,T), \ldots,\\ y(x_M,1), y(x_M,2), \ldots, y(x_M,T)]^T\end{aligned} \tag{3.7}$$

is the observed denoised spectra vector.

$$\lambda^{(q)} \triangleq [\lambda_1, \ldots, \lambda_k = q, \ldots, \lambda_N]^T,\ q \in \{0,1\}, \tag{3.8}$$

is an indicator vector for peptide existence.

$$\mathbf{\Sigma_e} = \text{diag}\left([\sigma_1^2, \ldots, \sigma_T^2; \sigma_1^2, \ldots, \sigma_T^2; \ldots; \sigma_1^2, \ldots, \sigma_T^2]_{1\times MT}\right), \tag{3.9}$$

with σ_t^2 being the variance of the t-th spectrum.

$$\mathbf{G} = (\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_N), \tag{3.10}$$

whose k-th column is given by

$$\begin{aligned}\mathbf{g}_k = [g_k(x_1,1), g_k(x_1,2), \ldots, g_k(x1,T), g_k(x_2,1), g_k(x_2,2), \ldots, g_k(x_2,T), \ldots,\\ g_k(x_M,1), g_k(x_M,2), \ldots, g_k(x_M,T)]^T,\end{aligned} \tag{3.11}$$

which is a $MT \times 1$ vector with the entry $g_k(x_m,t) = \sum_{i=1}^{cs}\sum_{j=0}^{iso} c_{k,ij} l_k(t) I_{x_m=\alpha_{k,ij}}$, $m = 1,2,\ldots,M$, $t = 1,2,\ldots,T$, representing the signal at (x_m,t) generated by peptide candidate k.

The apexes of the elution peaks of peptide candidate k at charge state i follow a multinomial distribution [52], which by the Central Limit Theorem can be approx-

imated by a Gaussian distribution as below:

$$\begin{aligned} P(c_{k,ij}, j = 0, \ldots, iso \,|a_k, \eta_{k,i}, \pi_k) &= MN(a_k \eta_{k,i}, \pi_k) && (3.12) \\ &\approx N(a_k \eta_{k,i} \pi_k, \\ &\quad a_k \eta_{k,i}[\mathrm{diag}(\pi_k) - \pi_k^T \pi_k]), && (3.13) \end{aligned}$$

where a_k is the total apex intensity of candidate k, $\eta_k \triangleq [\eta_{k,1}, \eta_{k,2}, \ldots, \eta_{k,cs}]^T$ denotes the candidate's charge state distribution, and $\pi_k \triangleq [\pi_{k,0}, \pi_{k,1}, \ldots, \pi_{k,iso}]^T$ is the theoretical isotopic distribution estimated by the Averagine approach [45, 56].

Thus the prior distribution of the apex vector $\mathbf{c}_k$ is given by:

$$\mathrm{Prior}(\mathbf{c}_k) = P(\mathbf{c}_k \,|a_k, \eta_k, \pi_k) \approx N(\mu_{\mathbf{c}_k}, \mathbf{\Sigma}_{\mathbf{c}_k}), \tag{3.14}$$

where

$$\begin{aligned} \mu_{\mathbf{c}_k} &= [a_k \eta_{k,1} \pi_k^T, a_k \eta_{k,2} \pi_k^T, \ldots, a_k \eta_{k,cs} \pi_k^T]^T, && (3.15) \\ \mathbf{\Sigma}_{\mathbf{c}_k} &= \mathrm{diag}(\Sigma_i), && (3.16) \end{aligned}$$

with

$$\Sigma_i = a_k \eta_{k,i}[\mathrm{diag}(\pi_k) - \pi_k^T \pi_k],\ i = 1, 2, \ldots, cs. \tag{3.17}$$

Substituting Eq. 3.6 and Eq. 3.14 into Eq. 3.5 and it can be shown by algebraic manipulations [53] that the conditional posterior distribution of $\mathbf{c}_k$ is also Gaussian, with the mean vector and covariance matrix given below:

$$\begin{aligned} \mathbf{\Sigma}_{\mathbf{c}_k|\mathbf{y},\theta_{-\mathbf{c}_k}} &= (\mathbf{I} - \mathbf{K}\mathbf{H}_k)\mathbf{\Sigma}_{\mathbf{c}_k}, && (3.18) \\ \mu_{\mathbf{c}_k|\mathbf{y},\theta_{-\mathbf{c}_k}} &= \mu_{\mathbf{c}_k} + \mathbf{K}(\mathbf{y} - \mathbf{G}\lambda^{(0)} - \mathbf{H}_k \mu_{\mathbf{c}_k}), && (3.19) \end{aligned}$$

where $\mathbf{H}_k = [h_{ms,(i-1)\times(iso+1)+j+1}]_{MT \times cs(iso+1)}$ is the elution profile matrix of candidate

k. The $[(i-1)\times(iso+1)+j+1]$th column contains the normalized elution profile of candidate k at charge state i and isotopic number j which has been estimated in preprocessing steps. And $\mathbf{K} \triangleq \mathbf{\Sigma}_{\mathbf{c}_k}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{\Sigma}_{\mathbf{c}_k}\mathbf{H}_k^T+\mathbf{\Sigma}_e\right)^{-1}$ is known as the Kalman gain matrix [54].

Note that the matrices involved in the above equations have huge dimensions which make the calculation almost infeasible. Thus, to update each peptide's signal, the related matrices $\mathbf{K}, \mathbf{G}, \mathbf{H}, \mathbf{y}$ and $\mathbf{\Sigma}_e$ are restricted to the corresponding peptide signal regions. This does no harm to the calculation accuracy while dramatically increases the speed.

For the kth peptide candidate, its total apex intensity, a_k, and its charge state distribution, η_k, are updated from the corresponding conditional posterior distribution derived in a similar manner as done in the previous chapter.

b. Sampling the peptide existence indicator variable

Let λ_k denote the existence indicator variable for peptide k. Its conditional posterior distribution is given by

$$\begin{aligned} P(\lambda_k\,|\mathbf{y},\theta_{-\lambda_k}) &\propto P(\mathbf{y}\,|\theta)\mathrm{Prior}(\lambda_k) \\ &\propto \exp\{-\frac{1}{2}(\mathbf{y}-\mathbf{G}\lambda)^T\Sigma_e^{-1}(\mathbf{y}-\mathbf{G}\lambda)\}\mathrm{Prior}(\lambda_k), \end{aligned} \tag{3.20}$$

where $\mathbf{G}$ is defined in Eq. 3.10.

The log-likelihood ratio (LLR) of λ_k can be calculated as below

$$\begin{aligned} LLR_{\lambda_k} &= \ln\frac{P(\lambda_k=1\,|\mathbf{y},\theta_{-\lambda_k})}{P(\lambda_k=0\,|\mathbf{y},\theta_{-\lambda_k})} \\ &= -\frac{1}{2}\left[(\mathbf{y}-\mathbf{G}\lambda^{(1)})^T\Sigma_e^{-1}(\mathbf{y}-\mathbf{G}\lambda^{(1)})-(\mathbf{y}-\mathbf{G}\lambda^{(0)})^T\Sigma_e^{-1}(\mathbf{y}-\mathbf{G}\lambda^{(0)})\right] \\ &\quad +\ln\frac{P(\lambda_k=1)}{P(\lambda_k=0)}, \end{aligned} \tag{3.21}$$

where $\lambda^{(q)}, q \in \{0, 1\}$ is defined by Eq. 3.8.

Absent prior knowledge about which peptide candidates are more likely to be present in the sample, then a reasonable choice is a uniform prior for λ_k. However, we wish to be conservative regarding the existence of peptide candidates. The idea is that by adding more candidates, it is possible to reduce the MSE between the inferred spectra and the observed denoised spectra, but at the same time the chances of overfitting increases as the model becomes more complex. Thus, a prior based on Bayesian information criterion (BIC) [61] is adopted to resolve the problem by introducing a penalty term for the number of parameters of the model. And the above equation can be rewritten as:

$$\begin{aligned} LLR_{\lambda_k} &= -\frac{1}{2}\left[(\mathbf{y}-\mathbf{G}\lambda^{(1)})^T\Sigma_e^{-1}(\mathbf{y}-\mathbf{G}\lambda^{(1)}) - (\mathbf{y}-\mathbf{G}\lambda^{(0)})^T\Sigma_e^{-1}(\mathbf{y}-\mathbf{G}\lambda^{(0)})\right] \\ &\quad -\frac{\ln(MT)}{2}\Delta, \end{aligned} \tag{3.22}$$

where $\Delta = Card(\theta) - Card(\theta_{-\lambda_k,-c_k}) = Card(\mathbf{c}_k)$ is the difference between the number of free parameters of the two models – with and without candidate k, respectively.

The conditional posterior distribution of λ_k is then obtained based on the log-likelihood ratio as given by Eq. 2.24 and 2.25.

For Gibbs sampling, it is well known that the correlation between parameters can reduce sampling efficiency. Thus, we cluster peptide candidates which have large overlaps in both m/z and RT dimensions together. Candidates within one cluster have strong correlations among each other, and their indicator variables are sampled from the joint conditional posterior distribution. The iteration order also affects the performance. Therefore, peptide clusters are first sorted by their importance, which is defined as the maximal intensity of the peptides in the cluster. The iteration starts from the most significant cluster and continues to the next significant one.

Our experimental results suggest that this scheme helps to reduce false positives and speed up convergence. The pseudocode of the entire Gibbs sampling process is given by Table IV. Samples taken after convergence can be used to estimate the target parameters. The existence probability of peptide k is calculated by Eq. 2.26.

If the LC-MS data also contain MS2 fragmentation spectra, then MS1 detected peptides can be linked to MS2 identified features given by software such as SEQUEST to obtain peptide sequence information.

C. Results and discussion

We report the observed performance of BPDA2d, side by side with state-of-the-art methods, such as msInspect and BPDA in a number of experiments using both synthetic and real data. The efficiency of BPDA2d in detecting low abundance and overlapping peptides is illustrated.

1. Results for synthetic data

a. Synthetic 100-mix LC-MS data sets with different abundance levels (SNRs)

First, to test robustness of various algorithms, we generated LC-MS data sets with different SNRs using methods described by [19]. More specifically, the mean signal strength (peptide abundance) was varied while the noise level (mean and variance of noise) was fixed. For each peptide abundance level $a \in \{100, 500, 5000\}$, the simulation was repeated 30 times. In each repetition, 100 true peptides (with abundance level a and masses randomly selected from tryptic digested human proteins) served as inputs of the model given by Eq. 3.4. The charge state distribution of one peptide was modeled by a binomial distribution, which was reported to approximate the real data well [19]. The isotopic distribution was calculated theoretically based on

Table IV. The Gibbs sampling process

1. Cluster candidates into S clusters.
2. Sort clusters by their importance in descending order.
3. For iteration $r = 1$ to R
4. For cluster $s = 1$ to S
5. For peptide candidate $k = i_1^s$ to $i_{N_s}^s$
6. Draw $\mathbf{c}_k^r$ based on its conditional posterior distribution.
7. end of k loop
8. Draw $\lambda_k^r, k = 1 \ldots, i_{N_s}^s$ for the cluster according to the joint conditional posterior distribution.
9. end of s loop
10. end of r loop

peptide sequence. The peptide elution profile was modeled by an exponentially modified Gaussian distribution, which captures different distortions of elution peaks by considering tailing and fronting effects [62]. Each output data set consists of 100 MS spectra with mass resolution 15,000.

BPDA2d, BPDA and msInspect (the latest Build 613) were applied to the same data sets to give detection results. We mainly focus on the performance comparison between BPDA2d and its precursor BPDA, which was shown to outperform popular algorithms such as OpenMS (Version 1.6.0), Decon2LS and VIPER in [39]. We also include msInspect in the comparison since it is widely used and has been reported to outperform other algorithms [63] such as MZmine. To apply BPDA, we followed the procedure introduced in the original paper [39]: Peptide elution peaks were first detected along the RT dimension, and elution peaks with similar RT were grouped. Each group contains a series of consecutive spectra, which were then averaged to form

a mean spectrum. Each mean spectrum was analyzed by BPDA, and finally an overall feature list was produced. To apply msInspect, we first wrote each simulated data set into a text file with three columns specified by RT, m/z, and intensity. Next, the text file was converted to mzXML and then msInspect was applied to give detection results including detected features and their qualities. The input parameters of msInspect were set to enable the inclusion of as many reasonable features as possible ("minpeaks" and "maxkl" were set to 2 and 10, respectively).

When comparing BPDA2d to its precursor BPDA, we found that the former had several advantages over the latter as detailed below. (I) For each experiment conducted, the total number of candidates considered in BPDA2d was greatly reduced compared to BPDA (reduced by 43% on average). This is expected since BPDA2d imposes additional constraints on candidates' elution peaks and can preclude non-reproducible noise peaks from the candidate list. To clarify, BPDA2d searches for candidates which can be repetitively observed across retention time — i.e. candidates whose elution peaks can be clearly identified. Thus, a major fraction of noise peaks (e.g. shot noise) which are not reproducible in time is removed. In contrast, BPDA is a 1D algorithm which works along the m/z dimension and processes one mean scan at a time. The mean scan is produced by taking the average of a few consecutive spectra. Thus, although noise in the form of random intensity fluctuation can be canceled out to some degree, non-reproducible noise peaks are still likely to be included in the resulting mean scan and therefore in the candidate list. Also, BPDA is likely to split long elution peaks into multiple mean scans and thus generate multiple candidates for a single true peptide. In summary, BPDA2d can compile a more reliable candidate list, which may help to reduce the number of detected false positives (FPs), and can allocate limited computational resources to candidates more likely to be true positives (TPs).

(II) BPDA2d reported significantly fewer FPs with existence probability larger than 0.9 than BPDA (reduced by 47% on average) and detected more TPs than the latter (increased by 6% on average). This improvement of BPDA2d is achieved by taking into account peptide elution peaks in addition to isotopic distribution and charge station distribution. BPDA2d tries to use all available observations from possible positions on the m/z-RT plane to infer the overall signal of each peptide candidate. By utilizing more information, detected signals become more reliable and the evidence of candidates' existence or non-existence becomes stronger, resulting in better detection results in terms of more TPs and less FPs.

When comparing BPDA2d to msInspect, we found that on average the TPs detected by BPDA2d increased by 16% than the latter while the FPs reduced by 40% (quality thresholds were set to existence probability larger than 0.9 and KL less than 1, for the two algorithms respectively).

To give a complete picture of the detection results, the classic Precision-Recall (PR) curve was adopted to evaluate the performance of various algorithms since the ground truth of the data is known. To obtain the PR curve, first a series of detection levels was selected, which range from the lower bound to the upper bound of feature quality scores (i.e. existence probability for BPDA and BPDA2d; KL score for msInspect). Features with quality score better than a given detection level were said to be detected at this specific detection level. A detected feature was claimed to be a true positive if it had the correct monoisotopic mass (e.g. within 10 ppm of the true mass), the correct RT (with a 6-scan tolerance), and the true RT is within the boundaries of the feature's elution peak ; otherwise the detected feature was called a false positive. Then, the True Positive Rate (TPR, i.e. recall) and precision (Prec) were calculated at each detection level as follows: $TPR = \frac{TruePositive}{TruePositive+FalseNegative}$ and $Prec = \frac{TruePositive}{TruePositive+FalsePositive}$. The averaged PR curve for one abundance level was

then obtained (each point on the curve was a pair of averaged precision and TPR at one detection level for all repetitions). We found that the Precision-Recall results were largely influenced by the size of the mass window used for matching detected features with the list of true peptides. Thus we plotted the Precision-Recall curves for various mass windows as shown in Fig. 4.

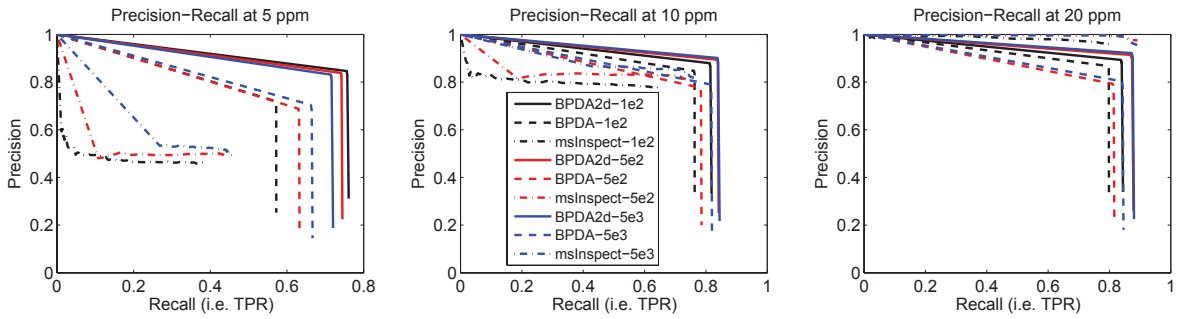


Fig. 4. Precession-Recall results for synthetic LC-MS data sets with different abundance levels (SNRs). Each panel shows the results obtained at a different mass window size as suggested by the title. Color codes for different abundance levels. Each method is represented by a unique line type. BPDA2d renders the best precision and sensitivity (i.e., recall) among all the methods compared for all abundance level in the first two mass window cases. In the last case, the performance between BPDA2D and msInspect has a very small difference.

In the PR space, the upper right corner (with a coordinate of [1,1]) represents 100% sensitivity (no false negatives) and 100% precision (no false positives). The closer the PR curve to the upper right corner, the better the algorithm. In this sense, BPDA2d is generally the best among all methods at all abundance levels. BPDA2d's performance is the least affected by the deterioration of SNRs among the three algorithms. Thus BPDA2d provides the most robust performance for lower abundance peptides.

Another advantage of BPDA2d is that it has much higher reported mass accuracy. Fig. 5 compares mass accuracies of all three algorithms. It can be seen that

the mass accuracy reported by BPDA2d is significantly higher than the other two algorithms. Given different mass accuracies, there is not a fair way for performance evaluation. Thus we provided performance evaluation in three cases when different mass window sizes were used. It can be seen that the mass window size does not affect the performance of BPDA2D significantly after 10 ppm because of its high mass accuracy. On the other hand, msInspect deteriorates quickly as we narrow the mass window from 20ppm to 10ppm due to its low mass accuracy. BPDA2D outperforms msInspect at higher mass accuracies of 10ppm and 5ppm. In the case of 20ppm, given the simple composition of the simulated data, the performance between BPDA2D and msInspect is similar. It shall be noted that with different mass accuracies by different algorithms, sample composition will strongly affect the reported PR curve — If the sample is more complex, with more peptides of similar weights, algorithm with lower mass accuracy like msInspect will further deteriorate in performance.

b. Synthetic LC-MS data set with 8 pairs of overlapping peptides

As noted, overlapping peptide peaks can complicate mass spectra and make the detection problem much harder. Hence, it is important to investigate algorithm performance in the presence of overlapping peptides. A synthetic 16-peptide-mix was generated by 8 pairs of overlapping tryptic digested human peptides. The data set contains 1000 LC-MS spectra with mass resolution 15,000. The intensity ratio of each pair (light/heavy) ranges from 0.25 to 3, and peptide charge states range from 1 to 4. More details on these peptides and the detection results of different algorithms are summarized in Table V.

For the first 4 pairs, the challenges are mainly to detect and split overlapping

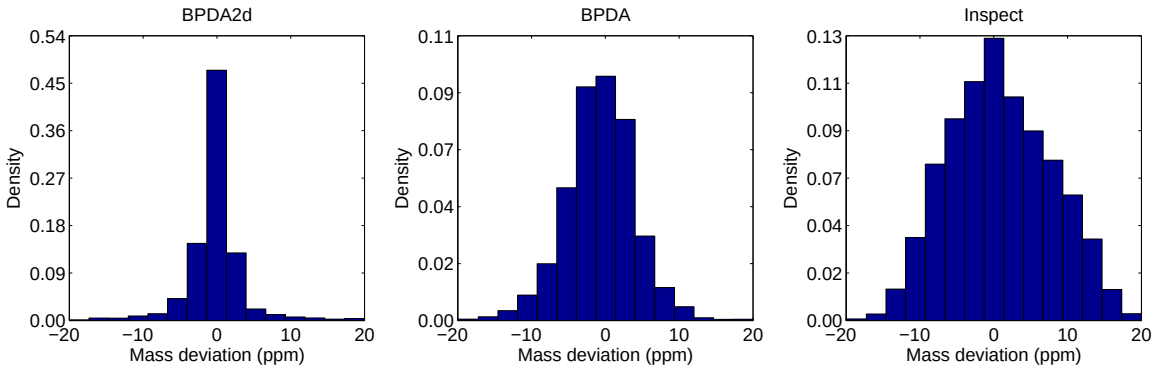
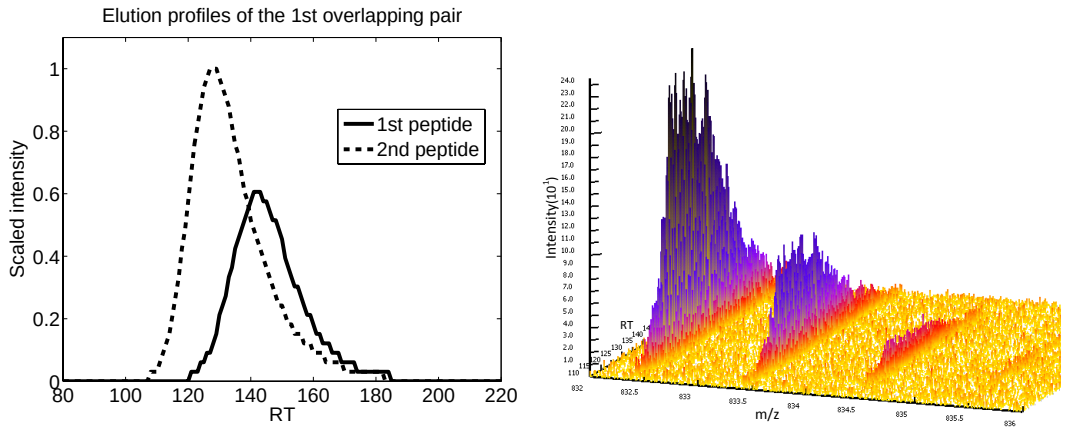


Fig. 5. Mass deviation of reported features that can be matched to the ground truth peptide list using a 20 ppm mass window (along with other criteria imposed on the retention time as described in the text). Each panel represents a detection algorithm as suggested by the subtitle. The plot was obtained by normalizing the mass deviation histogram by the total number of true peptides. It can be seen that BPDA2d has a much higher mass accuracy than the other two algorithms: the density around 0 ppm given by BPDA2d increased by around 4 times compared to BPDA and msInspect; and the SD of mass deviation is 3.7, 4.6, and 6.9 ppm for BPDA2d, BPDA and msInspect, respectively.

elution peaks of the two peptides in each pair with similar weights and close RT. For instance, the elution profiles and observed signals of the two peptides in the 1st pair are shown in Fig. 6. We observe that the two peptides have significant overlapping signal regions, which makes the detection problem tough. MsInspect experienced difficulty in identifying this pair. In fact, it failed to split the overlapping elution peaks and treated the two peptides as a single one. As a result, the intensity of the reported peptide (the 2nd one) equals the total intensity of the two. For BPDA, although it could report both peptides correctly, the intensity results were inaccurate (the intensity ratio turned out to be greater than 1 while the true ratio was 0.67). BPDA detected the second peptide correctly from 106s to 128s (approximately from the beginning to the maximum of the second peptide's elution profile, see Fig. 6(a)), while the rest of the signal peaks which appeared after 128s were shadowed by the



(a) (b)

Fig. 6. Overlapping signals of the first pair in 16-mix. a) Overlapping LC profiles of the two peptides. (b) Signal peaks of the two peptides at charge state 1 in a 3D view. SNR at this region is quite low, and significant peak overlapping can be observed.

1st peptide, whose signal was stronger. Therefore, in this region BPDA failed to include the 2nd peptide in its candidate list and tried to use the 1st peptide alone to explain the observed signal. The second peptide's corresponding intensity was thus wrongly attributed to the first one, thereby leading to inaccurate intensity results. In contrast to msInspect and BPDA, BPDA2d correctly split the elution peaks of the two peptides by capturing the tiny mass difference of the two and by detecting intensity dips in the observed overlapping peaks.

For the last 4 pairs, the weights of two peptides in each pair differ approximately by a multiple of the neutron weight. As a result, their isotopic peaks overlap at different isotope numbers and the overall isotope pattern deviates from each individual's. Thus, it is more challenging to utilize individual isotope pattern to discern the overlapping pair. As a vivid example, the elution profiles and the observed signals of peptides in the sixth pair are shown in Fig. 7. It is observed that the SNR at

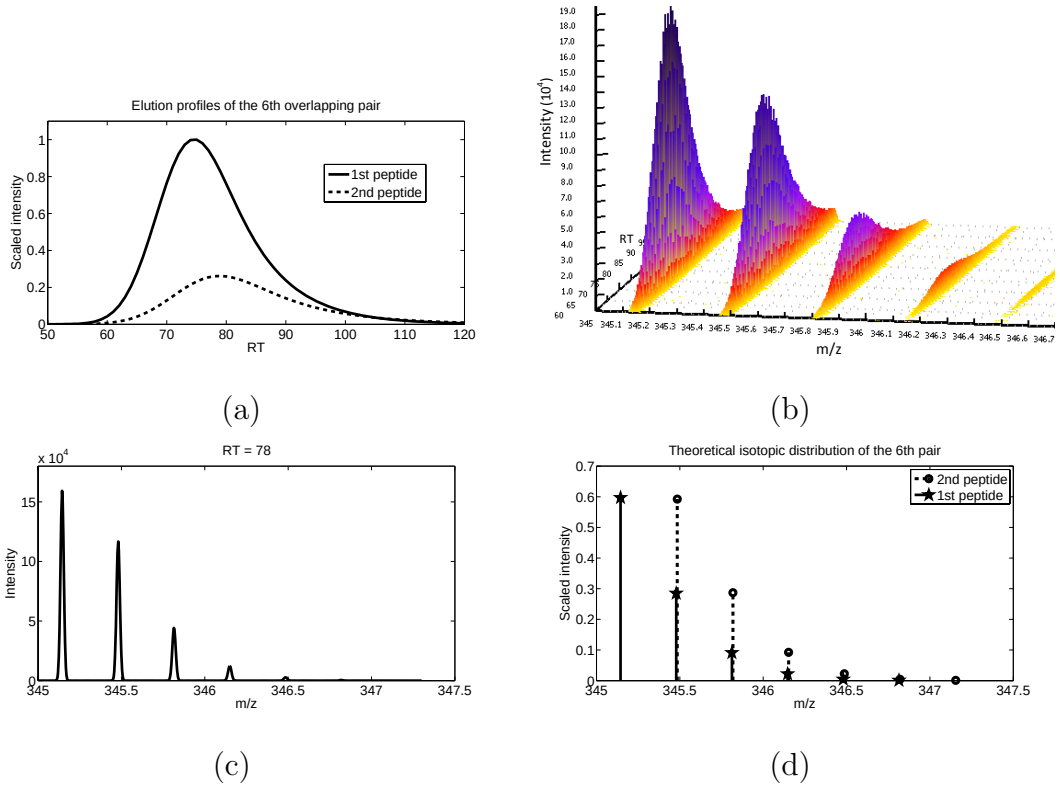


Fig. 7. Overlapping signals of the sixth pair in 16-mix. (a) Overlapping LC profiles. (b) Signal peaks of the two peptides at charge state 3 in a 3D view. This region has a high SNR, where peaks of the 2nd peptide almost get completely shadowed by all but the first isotope peak of the 1st peptide. (c) MS scan sampled at 78s showing signals of the same pair. The observed overall signal pattern deviates from (d) the theoretic isotope patterns of the two peptides.

Table V. Results of the data set with 8 pairs of overlapping peptides

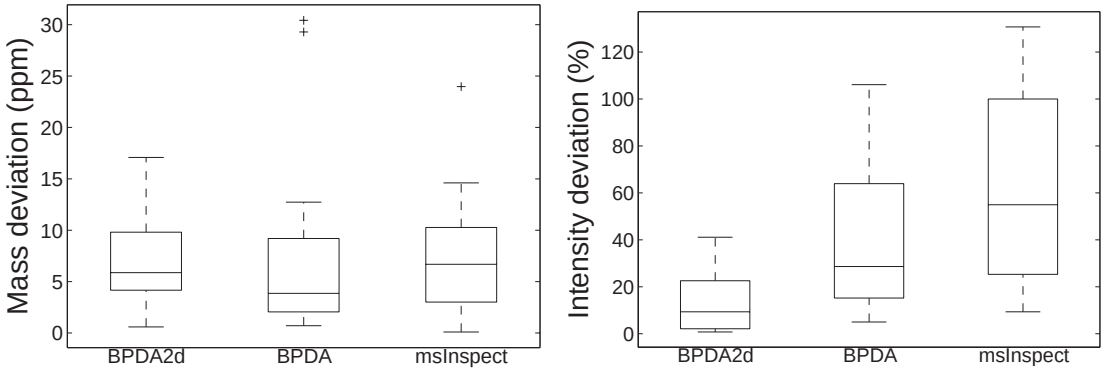
Pair No.	True peptide info					BPDA2d		BPDA		msInspect	
	Sequence	Mass(Da)	RT(s)	CS	Intn	CS	Intn	CS	Intn	CS	Intn
1	DYSYER	831.34	141	1-2	.0004	1-2	.0003	1-2	.0008	NA	
	DENGELR	831.37	127	1-2	.0006	1-2	.0006	1-2	.0005	1-2	.001
2	VVFMSLCK	925.48	414	1-2	.0046	1-2	.0033	1-2	.0043	1-2	.0050
	LLLPCLVR	925.58	456	1-2	.0054	1-2	.0044	1-2	.0056	1-2	.0068
3	MTPELMIK	961.50	323	1-3	.0001	1-3	.0001	1-3	.0001	1-2	.0001
	IAVMLMER	961.51	340	1-3	.0002	1-3	.0002	1-3	.0003	1-3	.0003
4	ACCLLCGCPK	1009.42	302	1-3	.0011	1-3	.0008	1-3	.0023	1-3	.0024
	MLCAGIMSGK	1009.48	314	1-3	.0008	1-3	.0009	1-3	.0014	1-3	.0020
5	AYDPDYER	1027.42	174	1-3	.0077	1-3	.0078	1-3	.0081	NA	
	EEPSGDGELP	1028.43	194	1-3	.0307	1-3	.0344	1-3	.0418	1-3	.0382
6	NGNEEGEER	1032.41	75	1-3	.4612	1-3	.5110	1-3	.7140	1-3	.7284
	TEGEEDAQR	1033.43	79	1-3	.1537	1-3	.1065	NA		NA	
7	MLANLVMHK	1055.56	312	1-3	.0019	1-3	.0018	1-3	.0017	1-3	.0023
	LTLDLMKPK	1057.62	321	1-3	.0009	1-3	.0010	1-3	.0008	NA	
8	LLPPLLQIVCK	1235.77	561	1-4	.1768	1-4	.1755	1-4	.1405	1-4	.1193
	LMLFMLAMNR	1238.63	577	1-4	.1537	1-4	.1516	1-4	.0779	1-4	.0943

CS and Intn denote detectable charge states and normalized intensity, respectively.

corresponding regions was quite high and peaks of the second peptide in this pair almost got completely shadowed under all but the first isotope peak of the first peptide (Fig. 7(a),(b)). Hence, the overall signal pattern (Fig. 7(c)) deviates from each individual's isotope pattern (see Fig. 7(d)). MsInspect was not able to detect this deviation: The calculated KL distance between the overlapping peak cluster and the first peptide's theoretical isotope pattern was surprisingly small (around 0.027), suggesting a 'good' match by its own criterion (a smaller KL score suggests a better match). MsInspect thus stopped there and assigned all overlapping signals to the first peptide, failing to consider the second peptide. This failure was not caused by chance. In fact, for the last 4 pairs, msInspect could correctly detect only one pair of peptides (the one with the least overlap) and missed one peptide in each of the other 3 pairs. This illustrates the inefficiency of template matching algorithms such as msInspect in dealing with overlapping isotope patterns as compared to BPDA2d and BPDA. Indeed, one should be wary of taking KL distance, or other distance

measures adopted by template matching algorithms, as a reliable measurement of the isotope pattern deviation. BPDA proposed a candidate corresponding to the second peptide; however, the candidate's existence probability was inferred to be 0, thereby rendering it undetectable. This was caused by the penalty term adopted in BPDA that penalizes model complexity. More specifically, the additional inclusion of the second peptide could reduce the MSE between the observed and inferred peaks to a small extent, but this reduction in MSE could not beat the increase of model complexity. Therefore, BPDA inferred the second peptide to be non-existent. Although BPDA2d utilizes a similar penalty strategy, the penalty term did not cause exclusion of the second peptide because BPDA2d used more observations from the m/z-RT plane than BPDA and the improvement in fitting the observed signal by inclusion of the second peptide was significant, thereby offsetting the penalty.

In summary, BPDA2d correctly detected all 46 charge states of the 16 peptides (along with 16 FPs), while BPDA and msInspect correctly detected 43 and 34 charge states, along with 57 and 4 FPs, respectively. All detected TPs of BPDA2d and BPDA had existence probability equaling to 1. For msInspect, the KL scores of TPs were less than 0.76. The box plots of mass and intensity deviation results given by the three algorithms are shown in Fig. 8. We observe that among the three algorithms, on average BPDA2d gave the most accurate abundance results and msInspect's results were the least accurate. BPDA had the best mass accuracy evaluated by the median mass deviation, but it rendered a few outliers and a larger variance compared to BPDA2d. Overall, msInspect produced the least accurate mass results. The synthetic test data are available upon request.



(a) (b)

Fig. 8. Box plots of (a) absolute mass deviation and (b) normalized intensity deviation of BPDA2d, BPDA and msInspect for the 16-mix data set

2. Results for real data

a. Data preparation

A QTOF LC-MS/MS data set was downloaded from the repository of the Seattle Proteome Center that is provided as a standard for testing algorithms. The data set was collected on a Waters/Micromass (Milford, MA) Q-TOF Ultima with Agilent 1100 series autosampler, Agilent 1100 series nanopump flowing at 200 nL/min and electrospray ionization. Approximately 200 fmol of total protein was injected on-column. The data set contains over 3500 MS1 spectra (m/z ranges from 250 to 1400 with FWHH around 0.15 Da.) and 775 MS2 spectra generated by peptides from 18 tryptic digested proteins (obtained from Bovine, Rabbit, Horse, etc.). More details can be found in [64]. MS1 level peptide detection was performed using BPDA2d, BPDA, and msInspect (the latest Build 613). We tried to optimize input parameters for msInspect: "minpeaks" was set to 2 and "maxkl" was set to 10, enabling the inclusion of as many reasonable features as possible. The "walksmooth" option was selected as it was recommended for QTOF data and improved the performance. For BPDA,

post-processing was applied to combine features that were split over consecutive mean spectra.

b. Comparative results

Direct comparison of results across different methods is meaningless unless ground truth of the data is known, but owing to contaminants and issue of peptide detectability, the true data composition is hard to know. As a workaround, SEQUEST and PeptideProphet were applied to analyze all the acquired MS2 spectra, rendering 234 unique peptide identifications associated with a high probability score (i.e. PeptideProphet score greater than 0.9) and could somehow reveal a portion of the truly existing peptides in the sample. We thus compared the detection results given by aforementioned MS1-based methods to the MS2 identifications. We say a MS1 feature is matched to a MS2 feature if the RT of the MS2 feature is within the retention peak of the MS1 feature and the mass deviation is within 40 ppm. The size of the mass window is chosen to include as many good matches as possible. It is larger than that used for synthetic data since here the ground truth peptide weights are unknown, and mass errors are associated with MS2 identifications as well as MS1 features. MS1 identifications were first filtered based on mass and RT. Only features with mass 1000-3710 Da. and RT in the range of 840 to 2030 scan were considered since all MS2 identifications were from these ranges. Remaining features were then selected based on the reported quality score. Because schemes used for calculating feature quality score vary across different algorithms, to ensure a fair and meaningful comparison, quality cutoff thresholds for various algorithms were carefully chosen as detailed below so that they corresponded to the same significance level.

- For BPDA2d and BPDA, existence probabilities are employed to measure fea-

ture quality. The cutoff thresholds of existence probability were calculated based on its null distribution, i.e. the distribution of existence probability of those candidates that are non-existing in the sample. We identify these peptides as those highly correlated (i.e. can be grouped into the same cluster as described in Method section) with one of the candidates that can be matched to the MS2 identification list. Although the ground truth is unknown, the latter candidates are likely to be TPs as they are confirmed by the MS2 identifications with high reliability, while the former are false identifications co-existing with the latter. These co-existing candidates should be assigned with a low existence probability. Given a significance level α, the corresponding threshold γ of the existence probability p can be calculated based on the right-tail probability of the null distribution: $\{\gamma | Prob(p \geq \gamma) = \alpha\}$.

- MsInspect uses KL score to measure feature quality. Cutoff KL thresholds were selected based on KL null distribution, i.e. the distribution of KL scores calculated between random noise and authentic isotopic distributions, as described in [65]. If KL score can faithfully reflect the deviation between random noise and real isotopic patterns, then the KL null distribution should skew to the right or have a small left-tail probability. On the other hand, given a significance level α, the corresponding KL threshold τ could be calculated based on the left-tail probability: $\{\tau | Prob(KL \leq \tau) = \alpha\}$.

From Fig. 9(a), it can be seen that BPDA2d detected many more features from the MS2 list than BPDA and msInspect at each significance level compared. Improvements are from 32% to 18%, and 64% to 19% compared to BPDA and msInspect, respectively, when significance level increases from 0.01 to 0.1, indicating a 3-6 fold increase in peptide coverage and quantification. In addition, all three MS1-

based algorithms detected significantly more features than that covered by the 234 MS2 identifications (see Fig. 9(b)), illustrating the under-sampling problem of MS2 and highlighting the benefits of employing MS1-based peptide detection algorithms to improve protein coverage rate. Performances of various algorithms were further investigated at a 0.05 significance level. The histogram of normalized intensity of MS2-level identifications detected by BPDA2d but not by msInspect is plotted in Fig. 9(c). The majority of identifications detected only by BPDA2d concentrate at the low intensity region (i.e. the area with normalized intensity less than 0.1), illustrating that BPDA2d can better identify low abundance peptides than msInspect. In addition, extra identifications yielded by BPDA2d did not cause degradation in mass accuracy (Fig. 9(d)). Moreover, BPDA2d slightly beat the other two methods in terms that the mean mass deviation is reduced by around 2%.

The average computational time of BPDA2d, BPDA and msInspect for testing data sets are 3.5 hr, 1 hr and 2.2 min, respectively. BPDA2d is expected to be more time-consuming since it looks for the optimal solution iteratively through Gibbs sampling on the whole spectra, while greedy template-matching based algorithms work on one local region at a time and calculate the fitting score, which typically does not require much computation. But we point out that the BPDA2D algorithm is fully parallelizable, and the authors are in fact working on a parallel version of the software that will be much faster.

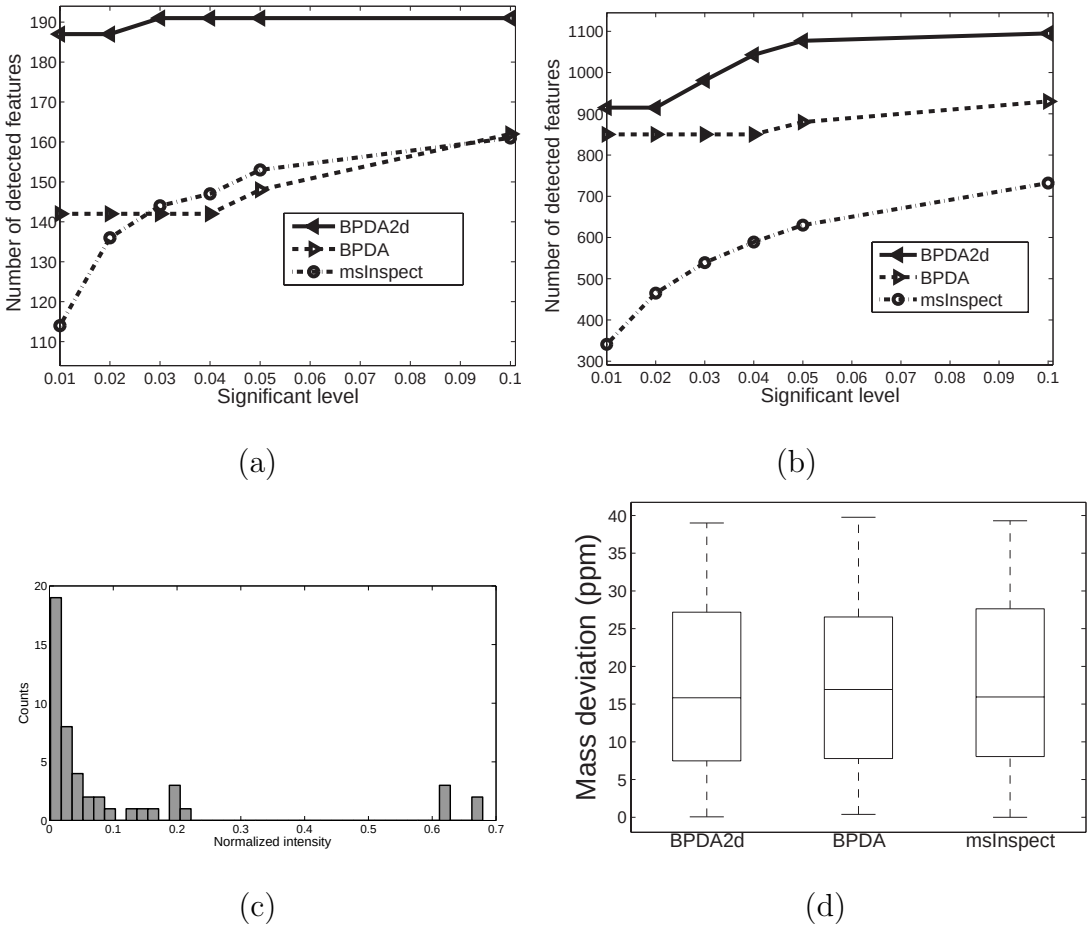


Fig. 9. Detection results of the QTOF LC-MS/MS data set. BPDA2d, BPDA and msInspect detected (a) number of features that can be matched to MS2 identifications at various significance levels and (b) total number of features. At significance level 0.05, the following two panels are obtained: (c) Histogram of normalized intensity of features detected by BPDA2d but not msInspect. Most of the features are from the low intensity region. (d) Box plots of absolute mass deviation of different algorithms.

CHAPTER IV

MODELING AND SYSTEMATIC ANALYSIS OF THE LC-MS PROTEOMICS PIPELINE

In this chapter, we model and evaluate the LC-MS data analysis pipeline from a systems point of view, with the goal of aiding experimental design, optimizing the workflow, predicting experiment results, and identifying key factors and bottlenecks that affect biomarker discovery results.

A. Background

1. Motivation

The MS analysis pipeline consists of many steps, including sample preparation, protein digestion, ionization, peptide detection, protein quantification, and so on. The pipeline can be viewed as a noisy channel, where each processing step introduces some loss or distortion to the underlying signal and the end results are affected by the combined effects of all upstream steps. While individual components of the MS pipeline have been studied at length, little work has been done to integrate the various modules, evaluate them in a systematic way, and focus on the impact of the various steps on the end results of differential analysis and sample classification. In real experiments, it is not easy to decouple the compound parameter effects and determine the marginal influence of various modules on the end results, due to variations and the complicated nature of the workflow. However, by employing a model-based approach, we may better understand the characteristics of the MS data, the contributions of the individual modules, and the performance of the full pipeline.

A key goal of MS-based proteomics is to discover protein biomarkers, which

can be used to improve diagnosis, guide targeted therapy, and monitor therapeutic response across a wide range of disease [6]. But to date, the rate of discovery of successful biomarkers is still unsatisfactory. Through the proposed model-based approach and by means of simulation using ground-truthed synthetic data, the problem of biomarker discovery can be studied and evaluated.

2. Results

In this work, we propose to model the Liquid Chromatography (LC) coupled MS system by identifying critical factors that influence system performance. Different modules are identified and integrated into the framework (see Fig. 10). The input of the pipeline can be any standard FASTA file containing proteins of interest. Here, we focus on analyzing protein drug targets downloaded from DrugBank [66], since LC-MS is an essential technology used to monitor these target proteins for drug development. We would like to point out that we are not trying to develop a detailed physical model for mass spectrometry as is, for instance, attempted in [49], which models the mass spectra generated by MALDI-TOF instruments. Rather, our purpose is to simulate the data flow realistically, but without descending into the physical parameters of the instrument itself. In addition, we do not focus only on MS data modeling, as done in [19], but we also address subsequent processes, including low level data analysis (e.g. peptide identification and quantification), and high level analysis (e.g. differential analysis and sample classification).

3. Application of the proposed model

The proposed LC-MS proteomic pipeline model can be used to determine the working range of important parameters and may shed light on experimental design. Also, if knowledge of sample complexity, instrument configuration, system variation and

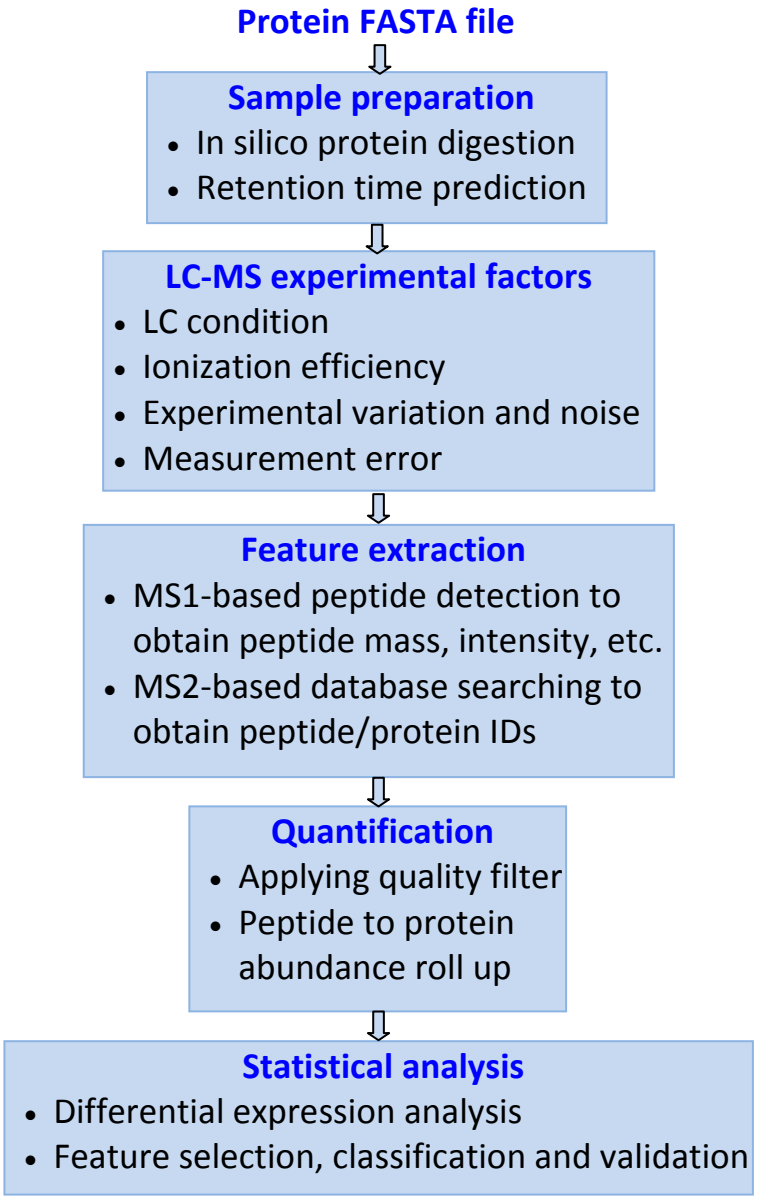


Fig. 10. The proposed MS-based proteomics pipeline.

detection accuracy is known beforehand, then by tuning corresponding parameters to their estimated values, the pipeline can be used to predict results on protein identification rates, protein differential analysis, quantification accuracies and classification performance. These results can be used to assess the efficacy of biomarker discovery in MS data.

B. Methods

1. Protein mixture model

In a typical label-free MS experiment, two sample classes (e.g. control vs. treatment) are considered. Assume each class has M samples and all samples share up to N_{pro} possible protein species of a given proteome. Protein concentration in the pooled control sample is modeled by a Gamma distribution in accordance with the observations in [67]:

$$\eta_l \sim \text{Gamma}(t, \theta)\,, \quad l = 1, 2, \ldots, N_{pro}\,, \tag{4.1}$$

where $t = 2$ and $\theta = 1000$ are the shape and scale parameters. The concentration has a dynamic range of approximately 4 orders of magnitude representing typical real-world scenarios. For the pooled treatment sample, expression levels of some proteins (e.g. biomarkers) may differ from those in the control sample, which can be captured by fold change:

$$f_l = \begin{cases} a_l, & \text{if protein } l \text{ is over-expressed} \\ \frac{1}{a_l}, & \text{if protein } l \text{ is under-expressed} \\ 1, & \text{otherwise} \end{cases} \tag{4.2}$$

where the fold change parameter, $a_l > 1$, is sampled from a uniform distribution, as specified in the Results section.

Sample variation of each protein is modeled by a Gaussian distribution [68], with means η_l and $\eta_l f_l$ in the control and treatment sample classes, respectively. Considering the fact that protein expression levels are often correlated, the following multivariate Gaussian (MVG) distribution is appropriate to model the interactions among proteins and their concentrations:

$$c_{lj}^{pro} \sim \begin{cases} \text{MVG}\left([\eta_1, \eta_2, \ldots, \eta_{N_{pro}}], \Sigma\right), & j \in \text{ class 0} \\ \text{MVG}\left([\eta_1 f_1, \eta_2 f_2, \ldots, \eta_{N_{pro}} f_{N_{pro}}], \Sigma\right), & j \in \text{ class 1} \end{cases} \tag{4.3}$$

where the covariance matrix Σ has a block-diagonal structure — proteins within the same block (e.g. proteins belonging to the same pathway) are correlated with correlation coefficient ρ and proteins of different blocks are uncorrelated [69]:

$$\begin{aligned} \Sigma &= [\sigma_{lj}^2]_{N_{pro} \times N_{pro}}, \\ \sigma_{lj}^2 &= \sigma_{ll}\sigma_{jj} r_{lj}\,, \end{aligned} \tag{4.4}$$

where σ_{ll} is proportional to the control protein mean η_l by a constant factor ϕ_l (i.e., the coefficient of variation), and the correlation coefficient matrix is

$$R = [r_{lj}]_{N_{pro} \times N_{pro}} = \begin{bmatrix} R_\rho & 0 & \cdots & 0 \\ 0 & R_\rho & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_\rho \end{bmatrix},$$

where R_ρ is a $D \times D$ matrix with 1 on the diagonal and ρ elsewhere. The correlation ρ and block size D are tunable parameters, with values specified in the Results section.

2. Peptide mixture model

Before being analyzed by the MS instrument, proteins are usually digested into peptides. In the proposed simulation pipeline, *in-silico* tryptic digestion is performed,

and retention time of peptide products is predicted using the PNNL Protein Digestion Simulator (`http://omics.pnl.gov/software/ProteinDigestionSimulator.php`). Different protein species may share the same peptide sequence. Thus, the molar concentration of peptide species i in sample j is given by the following equation:

$$c_{ij}^{pep} = \sum_{k \in \Omega_i} c_{kj}^{pro}\,, \quad i = 1, 2, \ldots, N_{pep}, \;\; j = 1, 2, \ldots, 2M\,, \tag{4.5}$$

where the set Ω_i comprises all proteins sharing the peptide species i, and N_{pep} is the number of peptide species. The concentration c_{ij}^{pep} is represented by ion abundance in MS data. Thus, the expected abundance readout μ_{ij} of peptide species i in sample j can be modeled as

$$\mu_{ij} = c_{ij}^{pep}\, e_i\, \kappa, \tag{4.6}$$

where e_i is a peptide efficiency factor similar to the one used in [70], and κ is the MS instrument response factor converting the original analyte concentration to the output ion current signal. The parameter e_i is affected by many factors: first, various peptides differ in hydrophobicity, which mainly determines their efficiencies in passing through the liquid chromatography column. Then, upon entering the ionization chamber, peptides demonstrate great disparities in ionization efficiency, which is affected by sample complexity, peptide concentration and characteristics such as polarity of side chains, molecular bulkiness, and so on [71]. In addition, some amino acids at the N-terminal end of peptides have destabilizing effects that can reduce the efficiency factor. Although there are methods attempting to predict e_i [70], they often neglect the fact that peptide efficiency and expected peptide ion abundance depend not only on the underlying peptide, but also on the combinational effects of other peptides present (e.g., LC elution competition, ion competition and suppression). In reality, it is unfeasible to predict e_i for all possible peptide combinations. Thus, we

model e_i from a uniform distribution and evaluate a wide range of interval bounds in simulations — we are not really interested in the precise value of e_i, but rather we aim to examine how the dispersion of e_i affects subsequent analysis. As for the parameter κ, it can be estimated through calibration and is related to the efficiency by which molecules are converted into gas-phase ions, the efficiency by which ions are transferred through various stages of the mass spectrometer, and how well experiment conditions are optimized. For a typical MS instrument, its response is linear for three to five orders of magnitude [71]. At high analyte concentration, instrument response plateaus because of detector saturation, restricted amount of excess charge, or limited space for ionization, as depicted in Fig. 11. To account for instrument saturation, an upper limit, sat, is set for the expected abundance readout: $\mu_{ij} = \min(\mu_{ij}, sat)$.

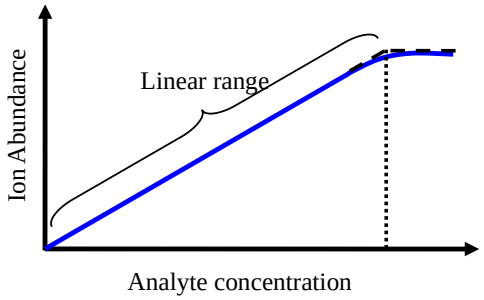


Fig. 11. The MS calibration curve which displays the MS ion signal as a function of analyte concentration in solution. The slope of the linear portion of the curve is the instrument response factor (i.e. instrument sensitivity). The curve departs from linear at high analyte concentration. A wider linear dynamic range is desired for quantitative analysis.

3. Peptide detection and identification

a. Peptide abundance

The actual abundance v_{ij} of peptide species i in sample j is modeled as the expected abundance plus Gaussian noise:

$$v_{ij} = \mu_{ij} + \epsilon_{ij}\,, \tag{4.7}$$

where

$$\epsilon_{ij} \sim \text{Gaussian}(0, \alpha\mu_{ij}^2 + \beta\mu_{ij})\,, \quad i = 1, 2, \ldots, N_{pep}\,, \;\; j = 1, 2, \ldots, 2M\,. \tag{4.8}$$

The sources of noise include variation in experimental conditions, instrument variance, thermal noise and measurement error. It is reported that the noise variance follows a quadratic dependence on the expected abundance [72], which is reflected by Eq. 4.8. The two parameters in the noise model, α and β, determine the noise severity. Their value can be estimated using replication analysis, as explained in [72].

In electrospray ionization, peptides can be multiply charged. But we do not model the charge distribution, considering the following facts: (1) Peptide charge distribution and the maximum charge states are complicated by many factors such as sample composition, analyte concentration and peptide conformation [73, 74]. The distribution is hard to predict and has not been well characterized. (2) In order to get the abundance of a peptide, and further, its parent protein, the abundance of peptide charge variants will eventually be summed up. We omit the intermediate process since in reality many factors involved are not well understood.

b. Peptide detection

Peptide detection from mass spectra is not an easy task — the observed peptide signals are corrupted by noise and may also be affected by signals of other peptides, and thus may deviate significantly from the expected pattern. The performance of a peptide detection algorithm on a specific MS instrument and the underlying SNR ultimately affect the number of detected true positives, i.e., the true positive rate (TPR), as shown in [28, 39, 40, 75]. The SNR is defined as the ratio of signal power to noise power, i.e., $SNR \triangleq E[v]^2/\mathrm{Var}(v) = 1/(\alpha + \frac{\beta}{\mu})$, see Eqs. 4.7–4.8. It can be seen that SNR increases as signal strength μ increases. The relationship between TPR and SNR can be approximated by a polynomial function, for algorithms such as those in [39, 40, 75]:

$$TPR = k \times SNR^p + b\,, \tag{4.9}$$

where b represents the worst TPR when the SNR approaches zero.

Besides SNR, signal interference and mass resolving power may also have considerable impact on TPR [19,40]. Over the years, much effort has been made towards enhancing instrument resolution, leading to improved mass accuracy, better separated MS peaks, and less convoluted peptide signals. But for complex samples, substantial overlapping of peptide signals is still frequently encountered, due to peptide isoforms or co-elution. It has been reported that if two peptides have overlapping signal regions, some detection algorithms may fail to report one of them even when the underlying SNRs are high, while other algorithms are shown to be superior in the detection of overlapping peptides [39]. To account for signal interference, we modify Eq. 4.9 by introducing an overlapping factor o_{ij}, so that the TPR of peptide species i in sample j becomes

$$TPR_{ij} = (k \times SNR_{ij}^p + b) \times o_{ij}\,, \quad o_{ij} \leq 1\,. \tag{4.10}$$

For algorithms such as NITPICK [28], BPDA [39] and BPDA2d [40], which are effective in detecting overlapping peptides, the overlapping factor o_{ij} can be approximated by 1, whereas for algorithms that are ineffective in detecting convoluted peptides, o_{ij} is assumed to be inversely proportional to the number of overlapping peptides, which is a function of the sample composition and the mass resolution. In our simulation, two peptide species i_1 and i_2 are said to overlap if their mass and retention time (RT) are close, in the sense that

$$\frac{|mass_2 - mass_1|}{mass_1} < \frac{1}{\text{mass resolution}} \quad \text{and} \quad \frac{|RT_1 - RT_2|}{\#\text{ scans}} < 0.005\,. \tag{4.11}$$

c. Peptide identification

The output of the MS1-based peptide detection algorithm is a list of detected peptides annotated by monoisotopic mass, retention time, abundance, and so on. To obtain peptide sequence information, i.e. peptide identification, which can be used to infer the parent protein from which the peptide was digested, database searching is required. To do so, the acquired MS/MS (MS2) spectra are searched against a protein database containing theoretical MS2 spectra generated from *in-silico* digested peptide sequences by popular software such as SEQUEST [76] and Mascot [20].

Several machine learning methods have been proposed to predict the probability (i.e., identifiability) of a peptide being identified through MS2 database searching [68, 77]. These methods try to extract the common trends residing in peptide identifiability that can be explained by peptide sequence-specific properties. Their successful application may suggest that the peptide sequence largely affects the chance of a peptide getting selected for MS2 analysis, whether the peptide can be sufficiently fragmented, and the quality of its fragmentation spectra. In our simulation, the identifiability p_i of the true peptide species i is predicted by the APEX software [68],

trained on the human serum proteome [78], and whether peptide species i in sample j is identified or not through database searching is determined by the outcome of a Bernoulli trial with success rate p_i.

d. Linking of detection and identification results

For both MS1-based and MS2-based algorithms, sources of error exist that give rise to false positives (FPs). For the former, error sources include shot noise, abundance measurement error, signal interference, and so on. For the latter, co-eluting precursor ions, spectra matching ambiguity, or post-translational modifications may all lead to false identifications. By confronting the results of the two orthogonal algorithms (i.e., a feature is treated as a true positive if it is reported by both algorithms), dubious features reported by either algorithm can be filtered out.

4. High-level analysis

a. Peptide to protein abundance roll-up

As demonstrated in the previous sections, each step of the MS analysis pipeline introduces a degree of loss or distortion to the underlying true signal. Thus, "decoding" protein abundance from observed peptide abundance corrupted by noise is nontrivial. To reduce noise, three levels of filtering are applied: (1) only unique peptides that exist only in one protein of the analyzed proteome are kept; (2) peptides with large missing value rates (larger than 0.7) are filtered out, since low reproducibility may be a red flag for false identifications; (3) among the remaining peptides, those having sufficiently high correlations (larger than 0.6) with other peptides digested from the same protein are retained. The estimated abundance of protein l in sample j is then obtained by averaging the abundances of its children peptides that pass the previous

filters; if less than two peptides pass the filters, the estimated protein abundance is set to zero. The estimated protein concentration is calculated by dividing the estimated protein abundance by the instrument response factor κ.

Quantification accuracy can be assessed by the commonly adopted mean quantification error, defined by

$$qerr \triangleq \frac{\sum_{l=1}^{N_{pro}} \sum_{j=1}^{2M} |c_{lj}^{prot} - \hat{c}_{lj}^{prot}|/c_{lj}^{prot}}{2MN_{pro}}, \tag{4.12}$$

where c_{lj}^{prot} and $\hat{c}_{lj}^{prot}$ are the original and estimated concentrations of protein l in sample j, respectively.

b. Differential expression analysis

Differential expression analysis is performed via a two-sample t-test with equal sample size and variance. The t statistic (or t score) is calculated as below:

$$t_l \triangleq \frac{|m_l^1 - m_l^0|}{\sqrt{\frac{Var_l^1 + Var_l^0}{M}}}, \tag{4.13}$$

where the superscripts identify the two classes, and m_l and Var_l represent the estimated class mean and variance of the abundance of protein l, respectively. The standard 0.05 significance level is used to detect differentially expressed markers.

c. Feature selection and classification

In the simulation, t-test feature selection is first performed to reduce the data dimension, by selecting the top 20 differentially expressed features. Then two classifiers, namely K-nearest neighbor (KNN, K=3) and linear discriminant analysis (LDA) are trained using the observed protein expression data. Classification performance is validated by independent ground-truth (testing) data sets (each with 1000 samples,

generated from the same data model), and the classification error is recorded. In addition, the KNN and LDA classification error on the original protein data (before entering the MS analysis pipeline) is obtained using a similar approach. The latter may serve as a benchmark to gauge how much loss in classification performance the analysis pipeline has introduced.

C. Results

To illustrate the application of the proposed pipeline model, a FASTA file containing around 4000 drug targets (human proteins) was compiled from DrugBank [66], which serves as the underlying proteome to be studied. In each run, 500 background proteins along with 20 marker proteins are randomly selected from the proteome to serve as the input of the pipeline. For each experimental setting studied, the simulation is repeated 50 times. We are interested in the effects of various factors on quantification, differential analysis, and classification. The study should be carefully designed to minimize parameter confounding effects. Thus, while examining the effects of one parameter, we either fix the values of other parameters, or try to eliminate their effects. Parameter configurations are given in Table VI, unless otherwise mentioned.

1. Sample characteristics

a. Effect of peptide efficiency factor

Though the exact distribution of the peptide efficiency factor e_i is unknown, we evaluate a wide range of values and try to find the common trend. It can be seen from Fig. 12(a) that as the lower bound of e_i increases, the quantification error decreases. This is expected since more ions can be detected by the instrument and transmis-

Table VI. Proteomics pipeline model summary

Parameters	Default values
No. of classes	2
Sample size of each class	$M = 50$
Proteome	Homo sapiens
No. of marker proteins	20
No. of non-markers	500
Protein block size	$D = 2$
Protein block correlation	$\rho = 0.6$
Fold change	$a_l \sim \text{Unif}(1.5, 2)$
Instrument response	$\kappa = 5$
Instrument saturation effect	$sat = \text{Inf}$
Noise level	$\alpha = 0.03, \beta = 3.6$
Peptide efficiency factor	$e_i \sim \text{Unif}(0.1, 1)$
Peptide detection algorithm	$b = 0, k = .0016, p = 2$
No. of MS2 replicates	1

sion loss is reduced as efficiency increases. Fig. 12(b) suggests that the percentage of observed differentially expressed proteins is positively correlated with e_i: this may be explained by the fact that as e_i increases, fewer missing values occur at the peptide level, and more proteins can be quantified in more samples, as can be seen in Fig. 12(c), resulting in more markers being detected by the differential expression test. Fig. 12(d) shows that the additional detected markers help to improve classification accuracy by decreasing the classification error.

b. Effect of protein abundance

The distribution of in-solution protein abundance can affect various detection results [79]. While high-abundance proteins are easily detectable, low-abundance proteins are hard to detect since their signals are more likely to be buried in background

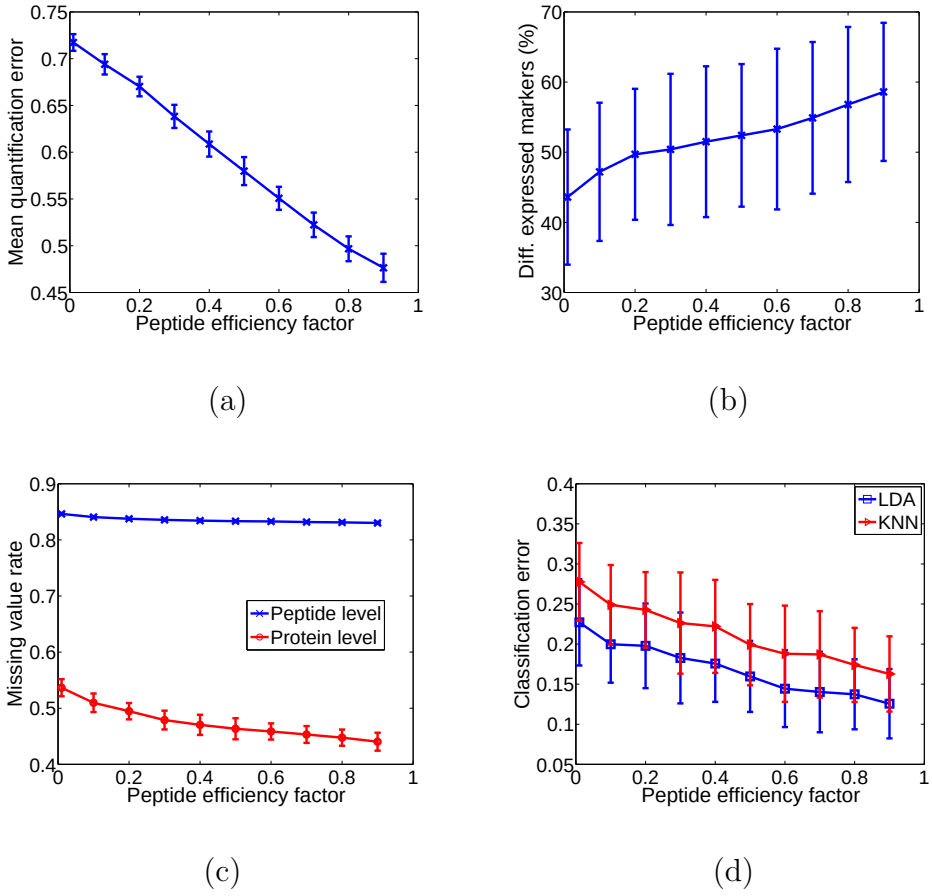


Fig. 12. Various quantities plotted as a function of the lower bound of peptide efficiency factor (the upper bound is fixed at 1). (a) Mean quantification error as defined in Eq. 4.12. (b) Percentage of observed differentially expressed marker proteins at a 0.05 significance level. (c) Missing value rates at the protein and peptide levels. (d) Classification error rates given by LDA and KNN classifiers, respectively.

noise. Hence, improving detection of low-abundance proteins has become a central issue in proteomic research.

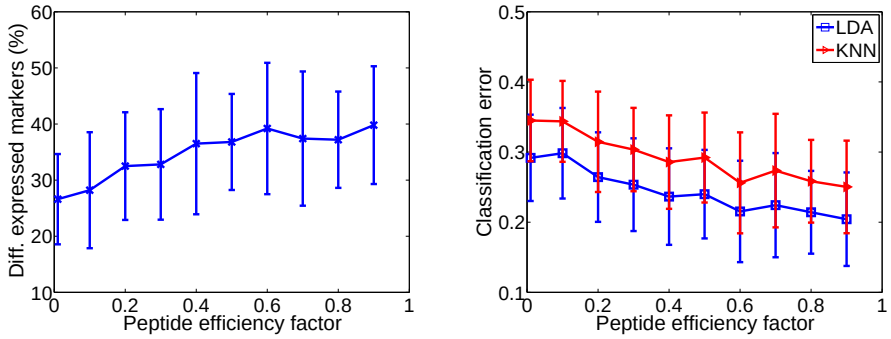
To demonstrate the effect of protein abundance on the detection of low-abundance marker proteins, we conduct an experiment where all markers are exclusively designed to have low abundance, distributed in the lower 25% quantile of the Gamma distribution; see Eq. 4.1. Fig. 13 depicts the corresponding plots to Fig. 12(b) and 12(d) in the case of the low-abundance markers. It can be observed that the percentage of detected differentially expressed markers and the classification results become worse compared to the results in Fig. 12(b) and 12(d). On average, the number of detected markers drops by 33.3% and the classification error increases by 42.4%. Similar trends are observed under other parameter settings (data not shown).

These results indicate that it is essential to develop methods to enhance the identification results of low abundance peptides which are often of more biological interests. Relative to hardware, sample fractionation and protein depletion through immunoaffinity-based approaches [80] can be helpful. Relative to software, there exist algorithms shown to be efficient for the detection of low-abundance peptides, such as BPDA2d [40].

c. Effect of sample size

Fig. 14 shows the effect of sample size. The range of values used is typical in proteomic experiments. It is observed that as more samples become available, the differential expression results and the classification accuracy improve notably. For instance, when sample size increases from 30 to 110, the number of detected markers increases by 41% and the classification error decreases by 40%.

In Fig. 14(b), the classification error of the (unobserved) original protein sample, before passing though the MS pipeline, is plotted side by side with that of the observed



(a) (b)

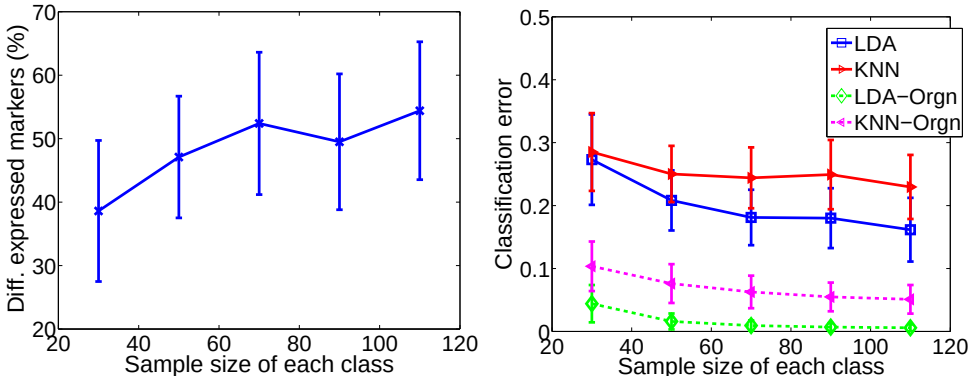
Fig. 13. Effect of peptide efficiency factor on (a) differential expression results, and (b) classification errors for samples with reduced marker concentration. Results deteriorate compared to those using the default protein concentration (Fig. 12(b) and 12(d)).

protein data, after analysis by the MS pipeline. The performance degradation caused by various noise conditions throughout the pipeline is clearly visible.

2. Instrument characteristics

a. Effect of instrument response

The effect of instrument response factor κ is displayed in Fig. 15. The experimental value of κ spans seven orders of magnitude. As κ first increases (from 0.1 to 100), true signals get amplified and SNRs become better, resulting in fewer missing values and false negatives at both peptide and protein levels (Fig. 15(a)), which in turn render better quantification and differential expression results (Figs. 15(b) and 15(c)). But when $\kappa > 100$, various performance indices level off. This illustrates that beyond a certain point, merely boosting the instrument response factor cannot help produce enhanced results. Rather, the performance bottleneck is determined by other factors such as noise in the system and efficiency of peptide detection algorithms.



(a) (b)

Fig. 14. Effect of sample size M on (a) differential expression results, and (b) classification error rates. All results generally improve as M increases. In (b) the classification error of the original protein sample (dashed lines) is plotted side by side with that of the observed protein data (solid lines), illustrating the substantial loss in accuracy introduced by the MS analysis pipeline.

b. Effect of saturation

In the previous experiment, the MS instrument is assumed to be working in the linear range. But for complex samples, for which analyte concentrations span orders of magnitude, saturation effects need to be taken into account (see Fig. 11). The previous experiment is repeated with the same settings, except that the saturation upper limit *sat* is changed from infinity to 10^4, corresponding to a 10^4 linear dynamic range when $\kappa = 1$. Interestingly, the resulting plots shown in Fig. 16 are no longer monotone as observed in Fig. 15. As the instrument response κ increases, the linear dynamic range (LDR) actually shrinks given the saturation ceiling is fixed (LDR can be approximated by sat/κ). Therefore, the percentage of peptides with saturated ion signals increases, and fewer peptides can pass the correlation filter, adversely affecting protein detection, quantification, and classification. To wit, when $\kappa > 10$, the protein missing value rate shoots up, fewer markers get detected, and classification

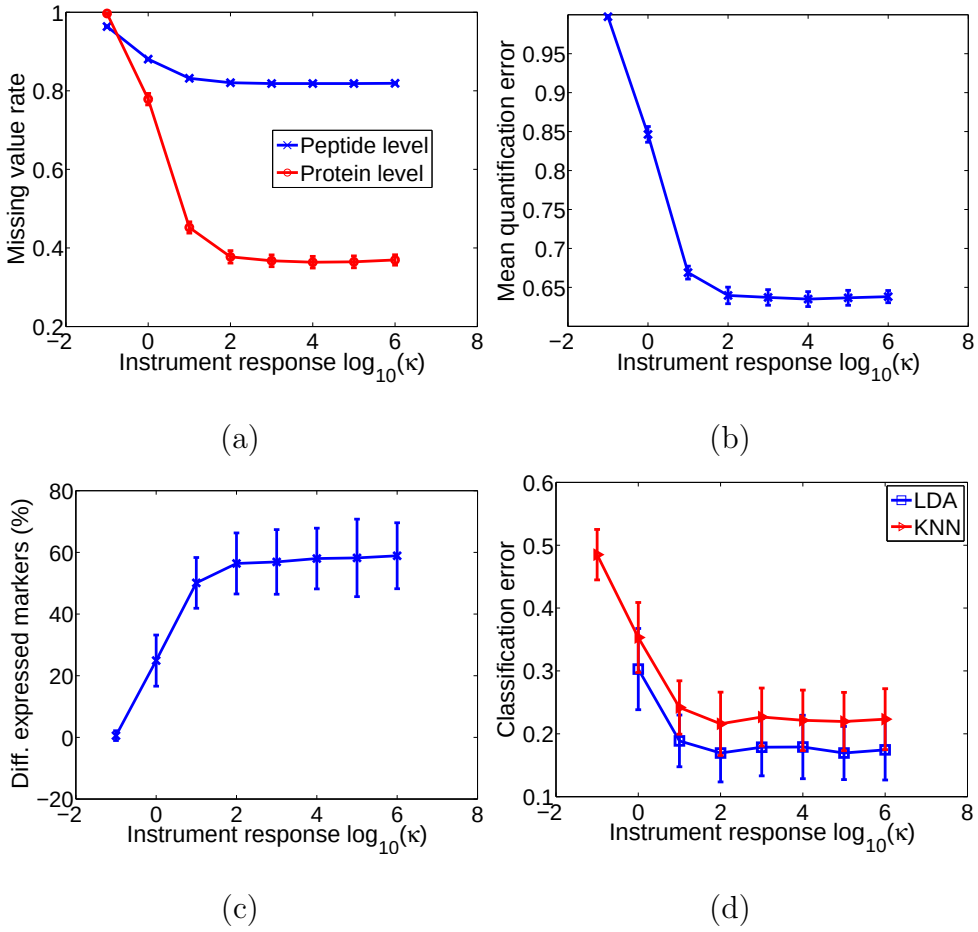


Fig. 15. Effect of instrument response factor κ on (a) missing value rates, (b) quantification accuracy, (c) differential expression results, and (d) classification error rates. As κ increases, all performance indices improve quickly and then level off.

performance and protein quantification results deteriorate.

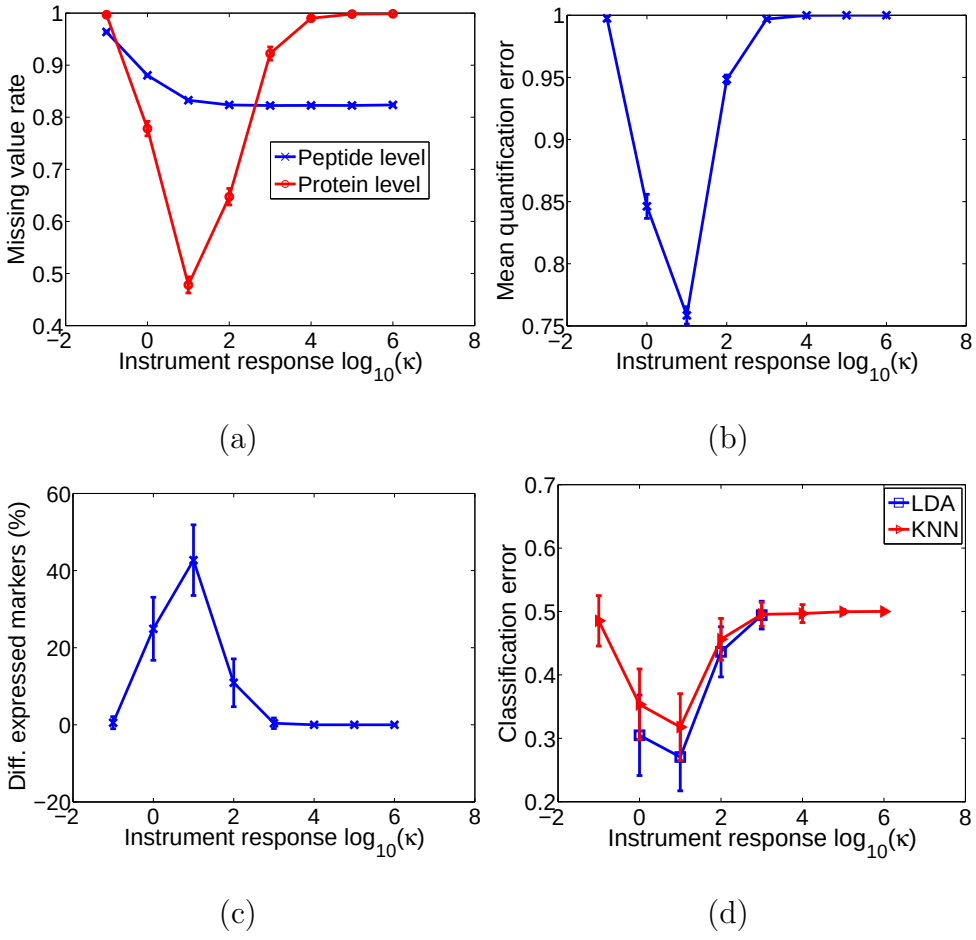


Fig. 16. Effect of instrument response κ in the presence of saturation on (a) missing value rates, (b) quantification accuracy, (c) differential expression results, and (d) classification error rates. As κ increases, all performance indices at first improve and then deteriorate (except for the peptide missing value rate, which levels off).

The compound effects of instrument sensitivity and saturation demonstrate that the effectiveness of MS in quantitative analysis relies on achieving a wide linear dynamic range with a high saturation ceiling and a matching sensitivity. For example, in electrospray ionization mass spectrometry, the linear range may be extended by enhancing gas-phase analyte charging, facilitating droplet evaporation, or introducing

ionization competitors [81].

c. Effect of noise

Noise in the MS analysis pipeline and the performance of peptide detection algorithms affect the number of proteins that can be quantified. To study noise impact directly, we eliminate the confounding effects of the peptide detection algorithm by assuming perfect detection, with $TPR \equiv 1$ for $SNR > 0$ and $TPR = 0$ for $SNR = 0$. It is observed in Fig. 17(a) that the peptide missing value rate stays relatively flat except at the end points where the accumulated effects of increasing noise levels are discernable: more of the true signal is obscured by noise and more peptides have infinitesimal SNR, which prevent their detection. The increasing trend in missing value rate at the protein level is more apparent: the fact that less proteins can be quantified as the noise level increases is not only due to fewer detectable peptides, but also because fewer peptides can pass the correlation filter for a protein to be quantified. Figures 17(b), 17(c) and 17(d) elucidate the adverse effects of noise on quantification accuracy, differential expression and classification results, respectively.

3. Peptide detection and experimental design characteristics

a. Effect of MS1 peptide detection algorithm

Given the same experimental settings, the performance of peptide detection algorithms may significantly affect the number of detected true positives (TPs). Three hypothetic detection algorithms with increasingly better performance are considered, in terms of TPR vs. signal strength curves; see Fig. 18(a). It can be seen in Fig. 18(b-e)) that the application of these detection algorithms leads to increasingly better re-

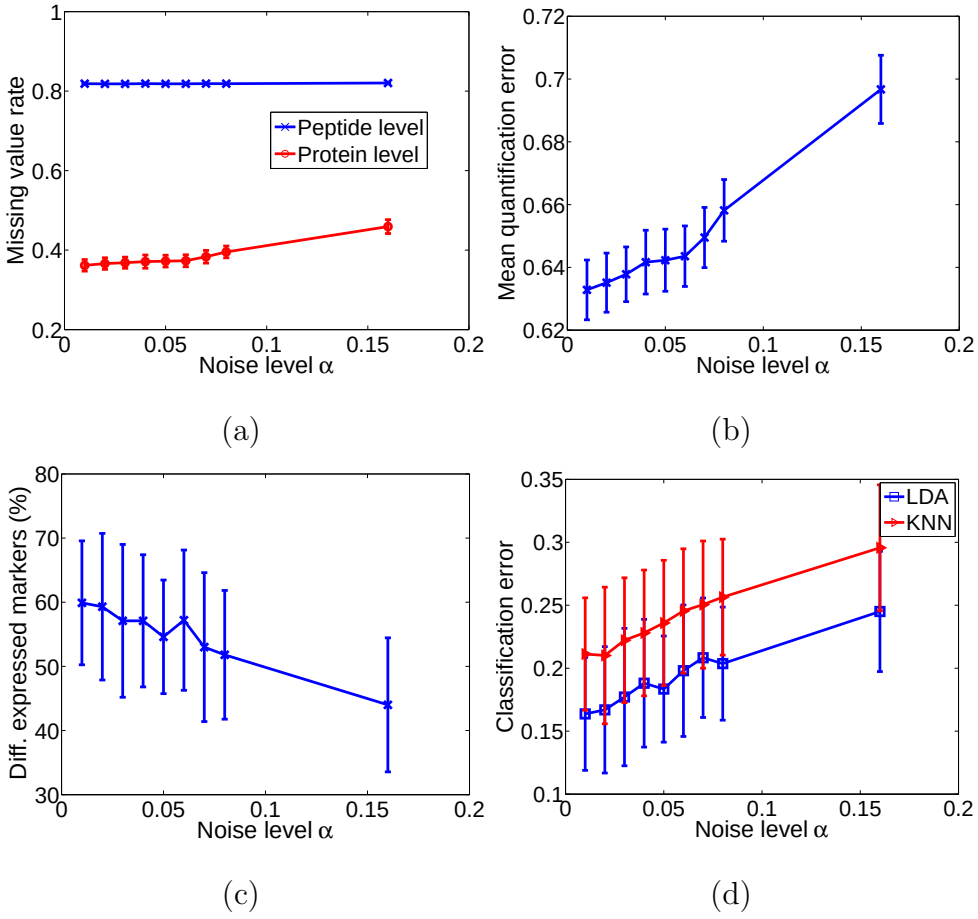


Fig. 17. Effect of noise on (a) missing value rates, (b) quantification accuracy, (c) differential expression results, and (d) classification error rates. The x-axis represents α in the noise model given by Eqs. 4.7–4.8, while β is set to be 120α. The parameter values in the middle of the range ($\alpha = 0.04, \beta = 4.8$) were estimated by an LC-MS analysis of human serum samples

sults in terms of missing value rate, quantification accuracy, detectable markers, and classification performance.

b. Effect of overlapping peptides and mass resolving power

To quantitatively evaluate the performance of MS1-based peptide detection algorithms under various mass resolutions and in the presence of overlapping peptides, two categories of detection algorithms are compared: the first characterizes those which can effectively detect convoluted peptides, such as NITPICK [28], BPDA [39] and BPDA2d [40], which are modeled by an overlapping factor $o_{ij} = 1$ in Eq. 4.10, and the second represents those that are sensitive to mass resolution and ineffective in detecting overlapping peptides (e.g. algorithms based on greedy template-matching), which are modeled by letting o_{ij} be inversely proportional to the number of overlapping peptides with peptide i in sample j. For algorithms in the first category, robust performance is expected for a range of mass resolutions (data not shown). In contrast, for algorithms in the second category, various performance indices generally become worse as mass resolving power declines, since more peptides cannot be resolved and are lost in detection (see Fig. 19). Summing up, the superiority of the first category over the second will be more evident for complex samples with more proteins and co-eluting analytes analyzed by a MS instrument with limited mass resolution.

c. Effect of MS2 replication

In tandem MS analysis, the precursor ions selected for fragmentation have low reproducibility across runs, and only a subset of peptides present in the sample can be analyzed for each run; this problem is known variously as MS2 random sampling and MS2 under-sampling [82]. Hence, though laborious and costly, replicate MS2 measurements are frequently conducted for in-depth proteomic profiling or for

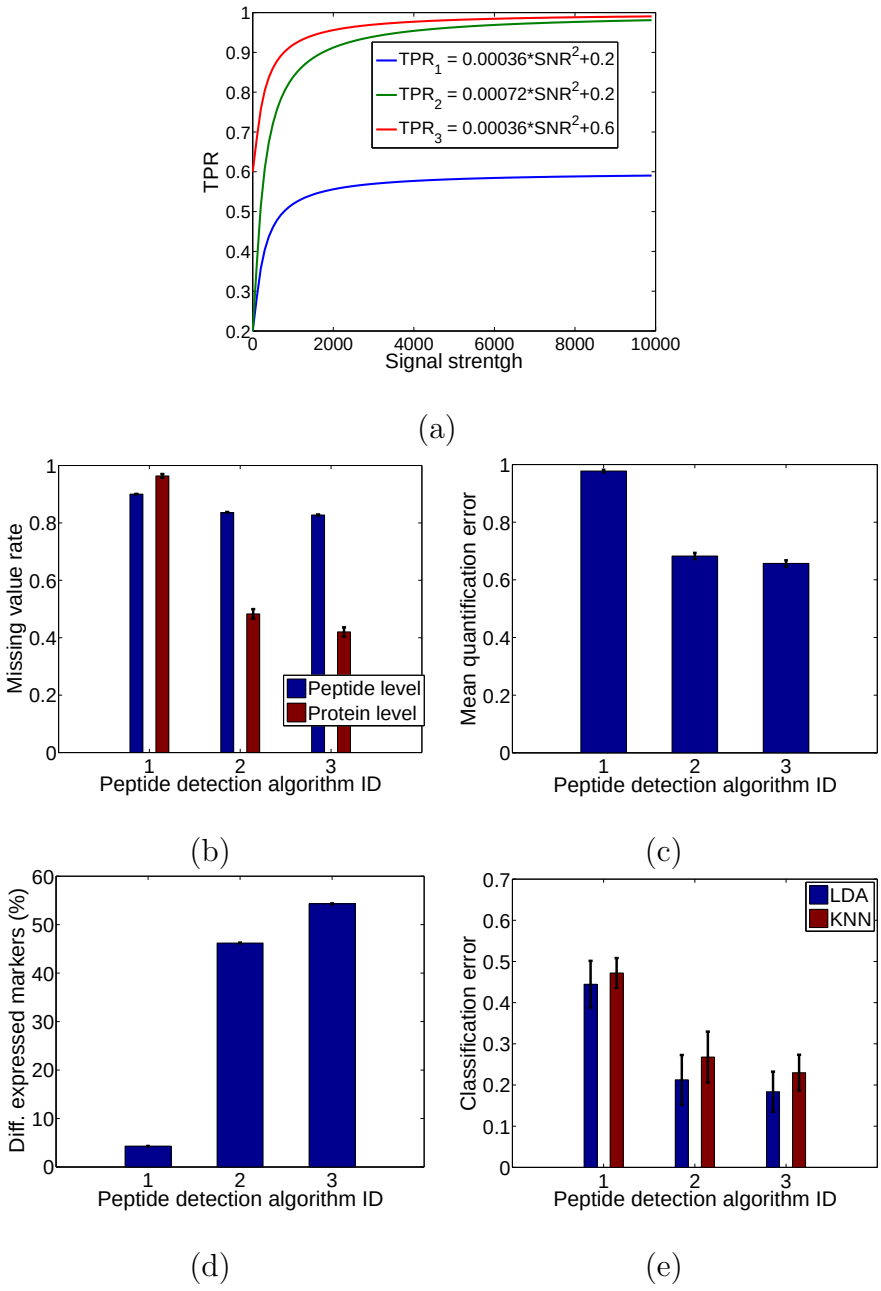


Fig. 18. Effect of using three hypothetic detection algorithms with increasingly better performance, quantified by the (a) TPR vs. signal strength curves. The applications of the three algorithms lead to increasingly improved results in terms of ((b) missing value rates, (c) quantification accuracy, (d) percentage of detectable markers, and (e) classification error rates.

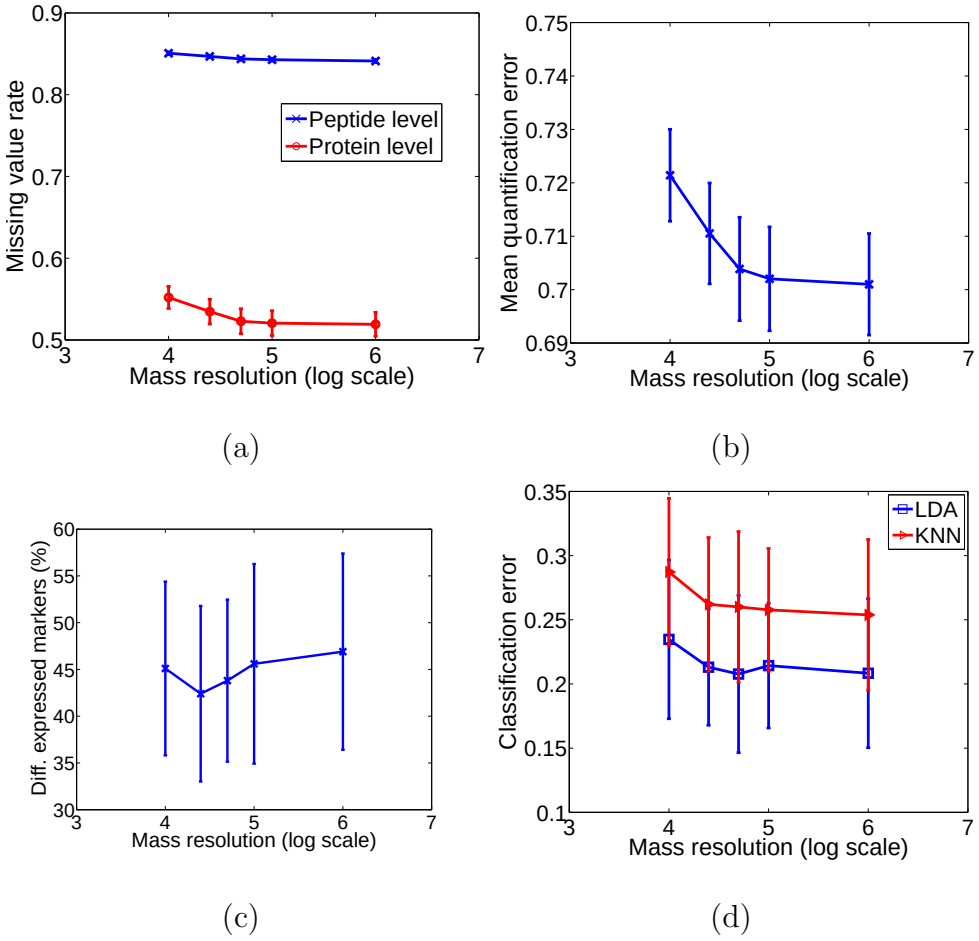


Fig. 19. Performance of a typical peptide detection algorithm in the second category described in the text under various mass resolutions and in the presence of overlapping peptides. (a) Missing value rates, (b) quantification accuracy, (c) differential expression results, and (d) classification errors.

building an AMT database to facilitate quantitative and high-throughput proteome measurements [83].

The effect of MS2 replication on various performance metrics is illustrated in Fig. 20. It is observed that even with a few replicate assays (as low as two or three), peptide and protein identification rates are boosted remarkably. As more replicates are made available, the protein identification rate levels off faster than the peptide rate, which was also observed in [78], indicating that newly identified peptides are mostly associated with already identified proteins. This may be explained as a bias towards relatively easily detectable proteins. Those proteins that are hard to detect may be a result of degradation, a spare amount of children peptides, ineffective ionization, and so on. Figs. 20(a) and 20(b) show that more proteins are detectable with improved quantification accuracy as the number of replicates increase. Comparing the use of three replicates against a single assay, Fig. 20(c) shows that the number of detected differentially-expressed marker proteins nearly doubles, while Fig. 20(d) indicates that the LDA classification error enjoys a 67% decrease.

4. Summary

The median value of each performance index across all previously studied cases with default sample size 100 is given in Table VII. It can be seen that the protein quantification rate exceeds the peptide identification rate. This may be explained by the one-to-many map from protein to its digested peptides: a protein can be quantified if more than one of its children peptides are identified and can pass the aforementioned quality filter. In the proteome studied, on average, one protein can be digested into around 20 peptides, and if we simply assume that each child peptide can be identified with a probability 0.17 (the calculated average peptide identification rate), independent of other peptides, and ignore the additional effects of the quality filter,

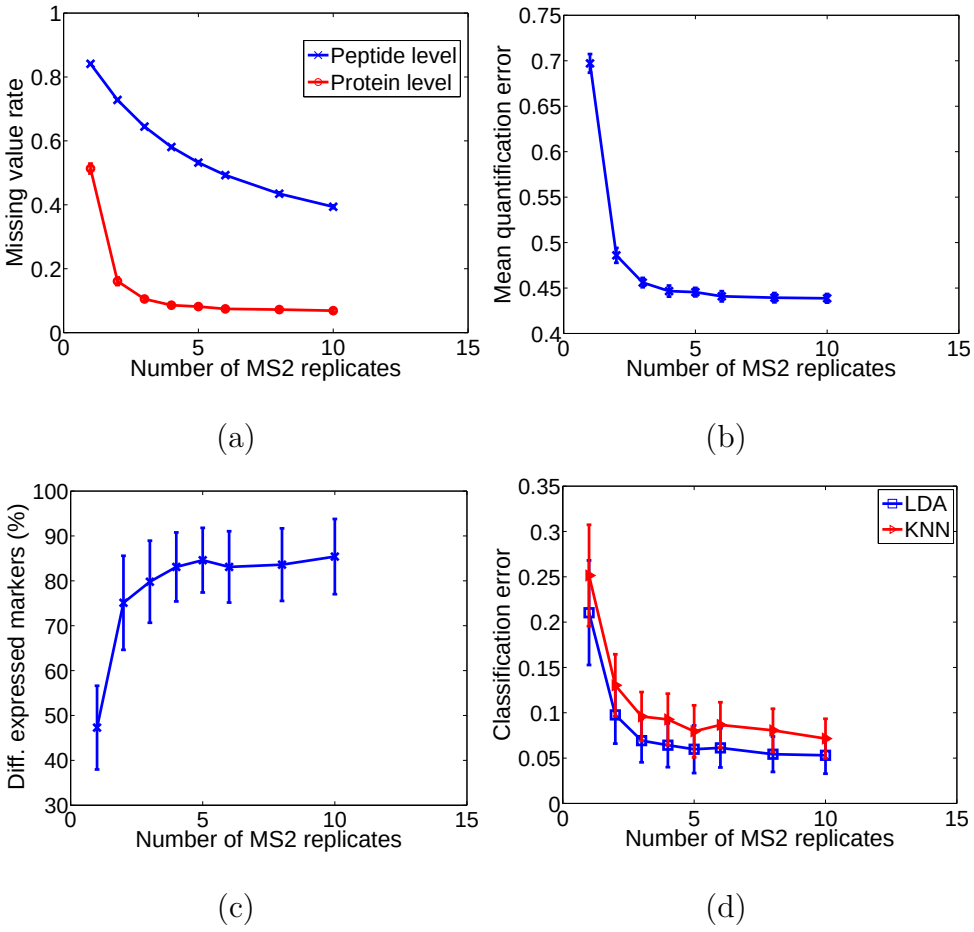


Fig. 20. Effect of MS2 replication on (a) missing value rates, (b) quantification accuracy, (c) differential expression results, and (d) classification errors. It can be seen that replicate analysis can significantly boost peptide and protein identification rates, quantification and classification results even only a few replicates are made available.

then the protein quantification probability (an upper bound) can be approximated by $1-(1-0.17)^{20}-20\times 0.17\times(1-0.17)^{19}=0.88$. The typical percentage of detected differentially-expressed protein markers is around 50% and the median value of the LDA classification error on the observed protein data is 0.18, which is 17 times larger than that of the original protein data — this exemplifies the signal corruption and error propagation introduced by the MS analysis pipeline, as well as the intricacy of biomarker discovery and their applications in disease diagnosis due to limited sample size, signal interference, ubiquitous noise, measurement errors, and so on.

Table VII. Results summary for the simulated MS-based proteomic pipeline

Performance indices	Median values
Peptide identification rate	0.17
Protein quantification rate	0.54
Protein quantification error	0.67
Percentage of detected markers	52%
LDA error on the original protein data	0.01
KNN error on the original protein data	0.03
LDA error on the observed protein data	0.18
KNN error on the observed protein data	0.24

D. Discussion

The main observations that were gleaned from the results of this study are as follows.

- Regarding sample characteristics, we observed a positive correlation between peptide efficiency and performance. The intricacy in detecting low-abundance peptides was demonstrated, thereby elucidating the advantage of sample fractionation and protein depletion through immunoaffinity-based approaches. More-

over, we showed that results could be improved by increasing sample size.

- As for instrument characteristics, the compound effects of instrument response and saturation were first examined and it was shown that the effectiveness of MS in quantitative analysis relies on achieving a wide linear dynamic range with a high saturation ceiling and matching instrument sensitivity. Enhancing gas-phase analyte charging, facilitating droplet evaporation, or introducing ionization competitors can be beneficial in extending the linear dynamic range. The adverse effects of noise was illustrated, highlighting the need in strictly following experiment protocols to minimize variance and measurement error.

- Peptide detection and experimental design characteristics were also studied. It was shown that improving peptide detection algorithms in the direction of enhancing true positive rate for a wide range of SNR (especially for low SNR) and tackling convoluted peptide signals could be invaluable, especially for complex samples and for MS instruments with limited mass resolution. It was also observed that the use of only a small number of replicate tandem MS assays could effectively reduce the MS2 under-sampling problem and improve performance.

To enable the performance analysis of such a complex system, many reasonable assumptions are made and the pipeline is simplified and reduced to a few key characteristics; nevertheless corruption of the true signal caused by the pipeline is evident and readily seen. This is expected to become worse as more steps are considered.

Though we used two sample types to illustrate the use of the LC-MS based pipeline model, the extension to multiple sample types is straightforward. In addition, the same methodology can be applied to study other MS platforms such as matrix-assisted laser desorption/ionization (MALDI). In addition, a similar strategy applies to labeled experiments.

CHAPTER V

MODEL-BASED STUDY OF MISSING VALUE IMPUTATION AND CLASSIFICATION IN DNA MICROARRAY DATA*

A. Introduction

Missing values (MVs) and low quality data points are frequently observed in microarray data. Many imputation methods have been proposed for estimating MVs in gene expression data which are usually organized in a matrix form with rows corresponding to the gene probes and columns representing the arrays. Trivial methods to deal with MVs in the microarray data matrix include replacing the MV by zero (given the data is in log domain) or by row average (RAVG). These methods do not make use of the underlying correlation structure of the data and thus often perform poorly in terms of estimation accuracy. Better imputation techniques have been developed to estimate the MVs by exploiting the observed data structure and expression pattern. These methods include K-nearest Neighbor imputation (KNNimpute) and singular value decomposition (SVD) based imputation [84], Bayesian principal components analysis (BPCA) [85], least square regression based imputation [86], local least squares imputation (LLS) [87], and LinCmb imputation [88], in which the MV is calculated by a convex combination of the estimates given by several existing imputation methods, namely, RAVG, KNNimpute, SVD and BPCA. In addition, a nonlinear PCA imputation based on neural networks was proposed for effectively dealing with nonlinearly structured microarray data [89]. Gene ontology based imputation utilizes

*Reprinted with permission from "Impact of missing value imputation on classification for DNA microarray gene-expression data: a model-based study" by Y. Sun, U. M. Braga-Neto, and E. R. Dougherty, *EURASIP Journal of Bioinformatics and Systems Biology*, vol.50, 17 pages, 2009, Copyright 2009 by SpringerOpen.

information on functional similarities to facilitate the selection of relevant genes for MV estimation [90]. Integrative MV estimation method (iMISS) aims at improving the MV estimation for data sets with limited numbers of samples by incorporating information from multiple microarray data sets [91].

In most of the studies about MV imputation, the performance of various imputation algorithms is compared in terms of the normalized root mean squared error (NRMSE) [84], which measures how close the imputed value is to the original value. However the problem is that the original value is unknown for the missing data, thus calculating NRMSE is infeasible in practice. To circumvent this problem, all the studies involving NRMSE calculation adopted the following scheme [84,86–88,91–93]: First, a sub-complete matrix is extracted from the original MV-contained gene expression matrix; Then, entries of the complete matrix are randomly removed to generate the artificial MVs; Finally, MV imputation is applied. The NRMSE can now be calculated to measure the imputation accuracy, since the original values are now known. This method is problematic for two reasons. First, the selection of artificial missing entries is random, and thus is independent of the data quality — whereas imputing data spots with low quality is the main scenario in real world. Secondly, in the calculation of the NRMSE, the imputed value is compared against the original, but the original is actually a noised version of the true signal value, and not the true value itself.

While much attention has been paid to the imputation accuracy measured by the NRMSE, a few studies have examined the effect of imputation on high-level analyses (such as biomarker identification, sample classification, and gene clustering), which demand that the data set be complete. For example, the effect of imputation on the selection of differentially expressed genes is examined in [88, 93, 94] and the effect of KNN imputation on hierarchical clustering is considered in [38], where it it

is shown that even a small portion of MVs can considerably decrease the stability of gene clusters and stability can be enhanced by applying KNN imputation. The effects of various MV imputation methods on the gene clusters produced by the K-means clustering algorithm are examined in [95], the main findings being that advanced imputation methods such as KNNimpute, BPCA and LLS yield similar clustering results, although the imputation accuracies are noticeably different in terms of NRMSE. To our knowledge, only two studies have investigated the relationship between MV imputation of microarray data and classification accuracy.

Wang *et al.* study the effects of MVs and their imputation on classification performance and report no significant difference in the classification accuracy results when KNNimpute, BPCA, or LLS are applied [96]. Five data sets are used: a lymphoma dataset with 20 samples, a breast cancer dataset with 59 samples, a gastric cancer dataset with 132 samples, a liver cancer dataset with 156 samples, and a prostate cancer dataset with 112 samples. The authors consider how differing amounts of MVs may affect classification accuracy for a given dataset, but rather than using the true MV rate, they use the MV rate threshold (MVthld) throughout their study, where, for a given MVthld (MVthld $= 5n\%$, where $n = 0, 1, 2, 4, 6, 8$), the genes with MV rate less then MVthld are retained to design the classifiers. As a result, the true MV rate (which is not reported) of the remaining genes does not equal MVthld and, in fact, can be much less than MVthld. Hence, the parameter MVthld may not be a good indicator. Moreover, the authors plot the classification accuracies against a number of values for MVthld, but as MVthld increases, the number of genes retained to design the classifier becomes larger and larger, so that the increase or decrease in the classification accuracy may be largely due to the additional included genes (especially if the genes are marker genes) and may only weakly depend on MVthld. This might explain the non-monotonicity and the lack of general trends in most of

the plots.

By studying two real cancer datasets (SRBCT dataset with 83 samples of 4 tumor types, GLIOMA dataset with 50 samples of 4 glioma types), Shi *et al.* report that the gaps between different imputation methods in terms of classification accuracy increase as the MV rate increases [97]. They test 5 imputation methods (RAVG, KNNimpute, SKNN, ILLS, BPCA), 4 filter-type feature selection methods (t-test, F-test, cluster-based t-test and cluster based F-test) and 2 classifiers(5NN and LSVM). They have two main findings: (1) when the MV rate is small ($\leq= 5\%$), all imputed datasets give similar classification accuracies that are close to that of the original complete dataset; however, the classification performances given by different datasets diverge as the MV rate increases; and (2) datasets imputed by advanced imputation methods (e.g. BPCA) can reach the same classification accuracy as the original dataset. A fundamental problem with their experimental design is that the MVs are randomly generated on the original complete dataset, which is extracted from the MV-contained gene expression matrix. Although this randomized MV generating scheme is widely used, it ignores the underlying data quality.

A critical problem within both aforementioned studies is that all training data and test data are imputed together before classifier design and cross-validation is adopted for the classification process. The test data influences the training data in the imputation stage and the influence is passed to the classifier design stage. Therefore, the test data is involved in the classification design process, which violates the principle of cross-validation.

In this paper, we carry out a model-based analysis to investigate how different properties of a dataset influence imputation and classification, and how imputation affects classification performance. We compare six popular imputation algorithms, namely RAVG, KNNimpute, LLS.L2, LLS.PC, LS and BPCA, by measuring how

well the imputed dataset can preserve the discriminant power residing in the original dataset. An empirical analysis using real data from cancer microarray studies is also carried out. In addition, the NRMSE-based comparison is included in the study, with a modification in the case of the synthetic data to give an accurate measure. Recommendations for the application of various imputations under different situations are given in the Results section.

B. Methods

1. Model for synthetic data

Many studies have shown the log-normal property of microarray data, that is, the distribution of log-transformed gene expression data approximates a normal distribution [98, 99]. In addition, biological effects which are generally assumed to be multiplicative in the linear scale become additive in the log scale, which simplifies data analysis. Thus, the ANOVA model [3, 100] is widely used, in which the log-transformed gene expression data are represented by a true signal plus multiple sources of additive noise.

There are other models proposed for gene expression data, including a multiplicative model for gene intensities [101]; a hierarchical model for normalized log ratios [102]; and a binary model [103]. The first two of these models do not take gene-gene correlation into account. In addition, the second model does not model the error sources. The binary model is too simplistic and not sufficient for the MV study in this paper.

Based on the log-normal property and inspired by ANOVA, we propose a model for the normalized log-ratio gene expression data which is centered at zero, assuming that any systematic dependencies of the log-ratio values on intensities have been

removed by methods such as Lowess [104, 105]. Here, we consider two experimental conditions for the microarray samples (e.g., mutant versus wild-type, diseased versus normal). The model can be easily extended to deal with multiple conditions as well.

Let X be the gene expression matrix with m genes (rows) and n array samples (columns). x_{ij} denotes the log-ratio of expression intensity of gene i in sample j to the intensity of the same gene in the baseline sample. x_{ij} consists of the true signal s_{ij} plus additive noise e_{ij}:

$$x_{ij} = s_{ij} + e_{ij}. \tag{5.1}$$

The true signal is given by

$$s_{ij} = r_{ij} + u_{ij}, \tag{5.2}$$

where r_{ij} represents the log-transformed fold change and u_{ij} is a term introduced to create correlation among the genes.

The log-transformed fold-change r_{ij} is given by:

$$r_{ij} = \begin{cases} a_i, & \text{if gene } i \text{ is up-regulated in sample } j, \\ 0, & \text{if gene } i \text{ is equal to the baseline in sample } j, \\ -b_i, & \text{if gene } i \text{ is down-regulated in sample } j, \end{cases} \tag{5.3}$$

under the constraint that r_{ij} is constant across all the samples in the same class. The parameters a_i and b_i are picked from a univariate Gaussian distribution, $a_i, b_i \sim \text{Normal}(\mu_r, \sigma_r^2)$, where the mean log-transformed fold change μ_r is set to 0.58, corresponding to a 1.5 fold change in the original linear scale, as this is a level of fold change that can be reliably detected [101]. The standard deviation of log-transformed fold change σ_r is set to 0.1.

The distribution of u_{ij} is multivariate Gaussian with mean 0 and covariance

matrix Σ. A block-based structure [69] is used for the covariance matrix to reflect the interactions among gene clusters. Genes within the same block (e.g. genes belong to the same pathway) are correlated with correlation coefficient ρ and genes within different blocks are uncorrelated as given by the following equation:

$$\Sigma = \sigma_u^2 \begin{bmatrix} \Sigma_\rho & 0 & \cdots & 0 \\ 0 & \Sigma_\rho & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_\rho \end{bmatrix}, \tag{5.4}$$

where

$$\Sigma_\rho = \begin{bmatrix} 1 & \rho & \cdots & \rho \\ \rho & 1 & \cdots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \cdots & 1 \end{bmatrix}_{D \times D}. \tag{5.5}$$

In the above equations, the gene block standard deviation σ_u, correlation ρ, and size D are tunable parameters, the values of which are specified in the Results section.

The additive noise e_{ij} in Eq. 5.1 is assumed to be zero-mean Gaussian, $e_{ij} \sim \text{Normal}(0, \sigma_i^2)$. The standard deviation σ_i varies from gene to gene and is drawn from an exponential distribution with mean μ_e to account for the non-homogeneous missing value distribution generally observed in real data [106]. The noise level μ_e is a tunable parameter, the value of which is specified in the Results section.

Following the model above, we generate synthetic gene expression datasets for the true signal, $\mathbf{S}$, and the observed expression values, $\mathbf{X}$. In addition, the dataset with MVs $\mathbf{X}^{MV}$ is generated by identifying and discarding the low quality entries of

$\mathbf{X}$, according to

$$x_{ij}^{MV} = \begin{cases} x_{ij}, & \text{if } |e_{ij}| < \tau \\ \text{MV}, & \text{o.w.} \end{cases} \tag{5.6}$$

The threshold τ is adjusted to give varying rates of missing values in the simulated dataset, as discussed in the Results section.

2. Imputation methods

Following the notation of [107], a gene with MVs to be estimated is called a target gene, with expression values across array samples denoted by the vector $\mathbf{y}_i$. The observable part and the missing part of $\mathbf{y}_i$ are denoted by $\mathbf{y}_i^{\text{obs}}$ and $\mathbf{y}_i^{\text{mis}}$, respectively. The set of genes used to estimate $\mathbf{y}_i^{\text{mis}}$ forms the candidate gene set $\mathbf{C}_i$ for $\mathbf{y}_i$. $\mathbf{C}_i$ is partitioned into $\mathbf{C}_i^{\text{mis}}$ and $\mathbf{C}_i^{\text{obs}}$ according to the observable and the missing indexes of $\mathbf{y}_i$. In row average imputation (RAVG), the MVs of the target gene $\mathbf{y}_i$ are simply replaced by the average of observed values, i.e. Mean($\mathbf{y}_i^{\text{obs}}$).

We will discuss three more complex methods, namely KNNimpute, LLS, and LS imputation, which follow the same two basic steps:

1) For each target gene $\mathbf{y}_i$, K genes with expression profiles most similar to the target gene are selected to form the candidate gene set $\mathbf{C}_i = [\mathbf{x}_{p_1}, \mathbf{x}_{p_2}, \cdots, \mathbf{x}_{p_K}]^T$.

2) The missing part of the target gene $\mathbf{y}_i^{\text{mis}}$ is estimated by a weighted combination of the corresponding K candidate genes $\mathbf{x}_{p_1}, \mathbf{x}_{p_2}, \cdots, \mathbf{x}_{p_K}$. The weights are calculated in different manners for different imputation methods.

We will additionally describe briefly the BPCA imputation method.

a. K-nearest neighbor imputation (KNNimpute)

In the first step, the L_2 norm is employed as the similarity measure for selecting the K neighbor genes (candidate genes). In the second step, the missing part of

the target gene ($\mathbf{y}_i^{\text{mis}}$) is estimated as a weighted average (convex combination) of the corresponding parts of the candidate genes ($\mathbf{x}_{p_l}^{\text{mis}}, l = 1, 2, \cdots, K$) which are not allowed to contain MVs at the same positions as the target gene:

$$\hat{\mathbf{y}}_i^{\text{mis}} = \sum_{l=1}^{K} w_l \, \mathbf{x}_{p_l}^{\text{mis}} . \tag{5.7}$$

The weight for each candidate gene is proportional to the reciprocal of the L_2 distance between the observable part of the target ($\mathbf{y}_i^{\text{obs}}$) and the corresponding part of the candidate ($\mathbf{x}_{p_l}^{\text{obs}}$):

$$w_l = \frac{f(\mathbf{y}_i^{\text{obs}}, \mathbf{x}_{p_l}^{\text{obs}})}{\sum_{l=1}^{K} f(\mathbf{y}_i^{\text{obs}}, \mathbf{x}_{p_l}^{\text{obs}})}, \tag{5.8}$$

where

$$f(\mathbf{y}_i^{\text{obs}}, \mathbf{x}_{p_l}^{\text{obs}}) = \frac{1}{\|\mathbf{y}_i^{\text{obs}} - \mathbf{x}_{p_l}^{\text{obs}}\|_2}, \quad l = 1, 2, \ldots, K. \tag{5.9}$$

The performance of KNNimpute is closely associated with the number of neighbors K used. A value of K within the range of 10-20 was empirically recommended, while the performance (in terms of NRMSE) degraded when K was either too small or too large [84]. We use the default value of $K = 10$ in the Results section.

b. Local least squares imputation (LLS)

In the first step, either the L_2 norm or the absolute value of the Pearson correlation coefficient is employed as the similarity measure for selecting the K candidate genes [87], resulting in two different imputation methods LLS.L2 and LLS.PC, respectively, with the former reported to perform slightly better than the latter. Owing to the similarity of performance, for clarity of presentation we only show LLS.L2 in the results section (the full results including LLS.PC are given on the companion website).

In the second step, the missing part of the target gene is estimated as a linear combination (which need not be a convex combination) of the corresponding parts of its candidate genes (whose MVs are initialized by RAVG):

$$\hat{\mathbf{y}}_i^{\text{mis}} \;=\; \sum_{l=1}^{K} w_l \, \mathbf{x}_{p_l}^{\text{mis}} \;=\; \left(\mathbf{C}_i^{\text{mis}}\right)^T \mathbf{w}\,, \tag{5.10}$$

where the vector of weights $\mathbf{w} = [w_1, w_2, \ldots, w_K]^T$ solves the least squares problem

$$\min_{\mathbf{w}} \left\| \left(\mathbf{C}_i^{\text{obs}}\right)^T \mathbf{w} - \mathbf{y}_i^{\text{obs}} \right\|_2 . \tag{5.11}$$

As is well-known, the solution is given by:

$$\mathbf{w} \;=\; \left(\left(\mathbf{C}_i^{\text{obs}}\right)^T \right)^{\dagger} \mathbf{y}_i^{\text{obs}}, \tag{5.12}$$

where $\mathbf{A}^{\dagger}$ denotes the pseudo-inverse of matrix $\mathbf{A}$.

c. Least squares imputation (LS)

In the first step, similar to LLS.PC, the K most correlated genes are selected based on their absolute correlation to the target gene [86].

In the second step, the least squares estimate of the target given each of the K candidate gene is obtained:

$$\hat{\mathbf{y}}_{i,l} = \bar{\mathbf{y}}_i + \beta_l(\mathbf{x}_{p_l} - \bar{\mathbf{x}}_{p_l}), \quad l = 1, \ldots, K\,, \tag{5.13}$$

where the regression coefficient β_l is given by

$$\beta_l \;=\; \frac{\text{cov}(\mathbf{y}_i, \mathbf{x}_{p_l})}{\text{var}(\mathbf{x}_{p_l})}, \tag{5.14}$$

where $\text{cov}(\mathbf{y}_i, \mathbf{x}_{p_l})$ denotes the sample covariance between the target $\mathbf{y}_i$ and the candidate $\mathbf{x}_{p_l}$ and $\text{var}(\mathbf{x}_{p_l})$ is the sample variance of the candidate $\mathbf{x}_{p_l}$.

The missing part of the target gene is then approximated by a convex combination of the K single regression estimates:

$$\hat{\mathbf{y}}_i^{\text{mis}} = \sum_{l=1}^{K} w_l \, \hat{\mathbf{y}}_{i,l}^{\text{mis}}, \tag{5.15}$$

The weight of each estimate is a function of the correlation between the target and the candidate gene:

$$c_l = \left(\frac{\text{corr}(\mathbf{y}_i, \mathbf{x}_{p_l})^2}{1 - \text{corr}(\mathbf{y}_i, \mathbf{x}_{p_l})^2 + 10^{-6}} \right)^2 \tag{5.16}$$

The normalized weights are then given by $w_l = c_l \Big/ \sum_{j=1}^{K} c_j$.

d. Bayesian principal component analysis (BPCA)

BPCA is built upon a probabilistic PCA model and employs a variational Bayes algorithm to iteratively estimate the posterior distribution for both the model parameters and the MVs until convergence. The algorithm consists of three primary processes, which are (1) principle component regression, (2) Bayesian estimation, and (3) an expectation-maximization-like repetitive algorithm [85]. The principal components of the gene expression covariance matrix are included in the model parameters, and redundant principal components can be automatically suppressed by using an automatic relevance determination (ARD) prior in the Bayes estimation. Therefore, there is no need to choose the number of principal components one want to use, and the algorithm is parameter free. We refer the reader to [85] for more details.

3. Experimental design

a. Synthetic data

Based on the previously described data model, we generate various synthetic microarray datasets by changing the values of the model parameters, corresponding to various noise levels, gene correlations, MV rates, and so on (more details are given in the Results Section). The MVs are determined by Eq. 5.6, with the threshold τ adjusted to give a desired MV rate. For each of the models, the simulation is repeated 150 times. In each repetition, according to Eq. 5.1 and Eq. 5.2, the true signal dataset, $\mathbf{S}$, and the measured-expression dataset, $\mathbf{X}$, are first generated. The dataset $\mathbf{X}^{MV}$ with missing values is then generated based on the data quality of $\mathbf{X}$ and a given MV rate. Next, six3 imputation algorithms, namely RAVG, KNNimpute, LLS.L2, LLS.PC, LS and BPCA are applied separately to calculate the MVs, yielding six imputed datasets, $\mathbf{X}_k$, for $k = 1, \ldots, 6$. Each of these training datasets contains m genes and n_r array samples and is used to train a number of classifiers separately. For each k, a measured-expression test dataset $\mathbf{U}$ and a missing value dataset $\mathbf{U}^{MV}$ are generated independently of, but in an identical fashion to, the datasets $\mathbf{X}$ and $\mathbf{X}^{MV}$, respectively. Each of these test sets contains m genes and n_t array samples, n_t being large in order to achieve a very precise estimate of the actual classification error.

A critical issue concerns the manner in which the test data are employed. As noted in the Introduction, imputation cannot be applied to the training and test data as a whole. Not only does this make the designed classifier dependent on the test data, it also does not reflect the manner in which the classifier will be employed. Testing involves a single new example, independent of the training data, being labeled by the designed classifier. Thus, error estimation proceeds in the following manner after imputation has been applied to the training data and a classifier designed from

the original and imputed values: (1) an example $U \in \mathbf{U}$ is selected and adjoined to the measured-expression training set $\mathbf{X}$; (2) missing values are generated to form the set $(\mathbf{X} \cup U)^{MV}$ [note that $(\mathbf{X} \cup U)^{MV} = \mathbf{X}^{MV} \cup U^{MV}$]; (3) imputation is applied to $(\mathbf{X} \cup U)^{MV}$, the purpose being to utilize the training data in the imputation for U^{MV} to obtain the complete vector U^{IMP} (the superscript IMP means one imputation method); (4) the designed classifier is applied to U^{IMP} and the error (0 or 1) recorded; (5) the procedure is repeated for all test points; and (6) the estimated error is the total number of errors divided by n_t. Notice that the training data are used in the imputation for the newly observed example, which is part of the classifier. The classifier consists of imputation for the newly observed example following by application of the classifier decision procedure, which has been designed on the training data, independently of the testing example. Overall, the classifier operates on the test example in a manner determined independently of the test example. If the imputation for the test data were independent of the training data, then one would not have to consider imputation as part of the classification rule; however, when the imputation for the test data is dependent on the training data, it must be considered part of the classification rule.

The classifier training process includes feature selection and classifier design based on a given classification rule. Three popular classification rules are used in this paper: Linear Discriminant Analysis (LDA), 3-Nearest Neighbor (3NN) and Linear Support Vector Machine (LSVM) [55]. Two feature selection methods, t-test and sequential forward floating search (SFFS) [108], are considered in our simulation study. The former is a typical *filter* method (i.e., it is classifier-independent) while the latter is a standard procedure used in the *wrapper* method (i.e., it is associated with classifier design and is thus classifier-specific). SFFS is a development of the sequential forward selection(SFS) method. Starting with an empty set A, SFS iter-

atively adds new features to A, so that the new set $A \cup \{f_a\}$ is the best (gives the lowest classification error) among all $A \cup \{f\}, f \notin A$. The problem with SFS is that a feature added to A early may not work well in combination with others but it cannot be removed from A. SFFS can mitigate the problem by "looking-back" for the features already in set A. A feature is removed from A if $A - \{f_r\}$ is the best among all $A - \{f\}, f \in A$, unless f_r, called the "least significant feature", is the most recently added feature. This exclusion continues, one feature at a time, as long as the feature set resulting from removal of the least significant feature is better than the feature set of the same size found earlier in the SFFS procedure [109]. For the wrapper method SFFS, we use bolstered error estimation [110]. In addition, considering the intense computation load requested by SFFS in the high-dimension problems such as microarray classification, a two-stage feature selection algorithm is adopted, in which the t-test is applied in the first stage to remove most of the noninformative features and then SFFS is used in the second stage [69]. This two-stage scheme takes advantage of both the filter method and the wrapper method and may even find a better feature subset than directly applying the wrapper method to the full feature set [111]. In summary, for each of the data models, 8 pairs of training and testing datasets are generated and are evaluated by a combination of 2 feature selection algorithms and 3 classification rules, resulting in a very large number of experiments.

Each experiment is repeated 150 times, and the average classification error is recorded. The averaged classification error plots for different datasets, feature selection methods and classification rules are shown in the Results section. Besides the classification errors, the NRMSE between the signal dataset and each of the 6 imputed datasets are also recorded. The simulation flow chart is shown in Fig. 21.

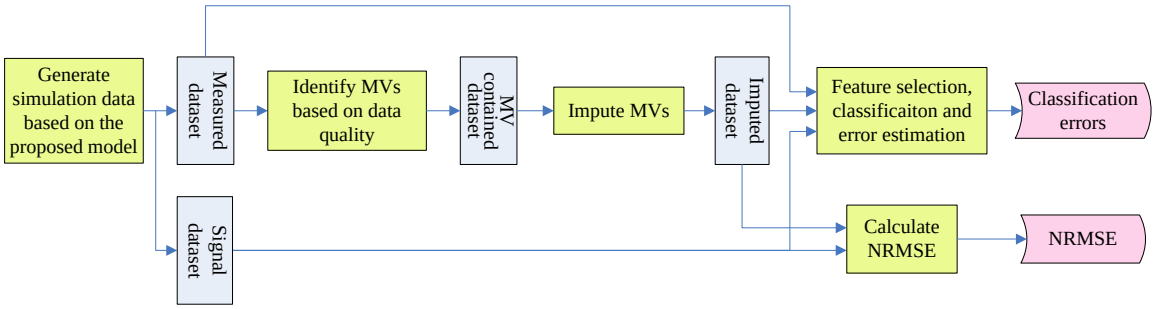


Fig. 21. Simulation flow chart

As previously mentioned, there can be drawbacks associated with the NRMSE calculation; however, in our simulation study, the MVs are marked according to the data quality and the NRMSE is calculated based on the true signal dataset which can serve as the ground truth:

$$\text{NRMSE} = \frac{\sqrt{\text{Mean}[(\mathbf{x}^{imputed} - \mathbf{x}^{true})^2]}}{\text{Std}(\mathbf{x}^{true})}.$$

In this way, the aforementioned drawbacks about using NRMSE are addressed.

b. Patient data

In addition to the synthetic data described in the previous section, we used the two following publicly-available data sets from published studies:

- **Breast cancer dataset (BREAST)**: Tumor samples from 295 patients with primary breast carcinomas were studied by using inkjet-synthesized oligonucleotide microarrays which contained 24,479 oligonucleotides probes along with 1281 control probes [112]. The samples are labeled into two groups [113] : 180 samples for poor-prognosis signature group, and 115 samples for good-prognosis signature. In addition to the log-ratio gene expression data, the log error data is also available which can be used to assess the data quality.

- **Prostate cancer dataset (PROST)**: Samples of 71 prostate tumors and 41 normal prostate tissues were studied, using cDNA microarray containing 26,260 different genes [114]. In addition to the log-ratio gene expression data, additional information such as background (foreground) intensities and SD of foreground and background pixel intensities are also available and thus can be used to calculate the log error according to the Rosetta error model [115] — the log error $e(i,j)$ for the i-th probe in the j-th microarray sample is given by the following equation:

$$e(i,j) \propto \sqrt{\frac{\sigma_1^2(i,j)}{I_1^2(i,j)} + \frac{\sigma_2^2(i,j)}{I_2^2(i,j)}} \tag{5.17}$$

where

$$\sigma_k^2(i,j) = \frac{\sigma_{k,fg}(i,j)^2}{N_{k,fg}(i,j)} + \frac{\sigma_{k,bg}(i,j)^2}{N_{k,bg}(i,j)} \tag{5.18}$$

and

$$I_k(i,j) = I_{k,fg}(i,j) - I_{k,bg}(i,j), \quad k = 1,2. \tag{5.19}$$

In the above equations, k specifies the red or green channel in the two-dye experiment, $\sigma_{k,fg}(i,j)$ and $\sigma_{k,bg}(i,j)$ denote the SD of foreground and background pixels, respectively, of the i-th probe in the j-th microarray sample, $N_{k,fg}$ and $N_{k,bg}$ are the numbers of pixels used in the mean foreground and background calculation, respectively, and $I_{k,fg}$ and $I_{k,bg}$ are the mean foreground and background intensities, respectively.

For the patient data study, the schemes used for imputation, feature selection and classification are similar to those applied in the synthetic data simulation, except

that we use hold-out-based error estimation, i.e. in each repetition, n_r samples are randomly chosen from all the samples as the training data and the remaining $n_t = n - n_r$ samples are used to test the trained classifiers, with n_t being much larger than n_r in order to make error estimation precise. We preprocess the data by removing genes which have an unknown or invalid data value in at least one sample (flagged manually and by the processing software). After this preprocessing step, the dataset is complete, with all data values being known. We further preprocess the data by filtering out genes whose expressions do not vary much across all the array samples [95] [114]; indeed, the genes with small expression variance do not have much discrimination power for classification and thus are unlikely to be selected by any feature selection algorithm [97]. The resulting feature sizes are 400 and 500 genes for the prostate and the breast dataset, respectively. It is at this point where we begin our experimental process by generating the MVs.

Unlike the synthetic study, the true signal dataset is unknown in the patient data study since the data values are always contaminated by measurement errors. Therefore, in the absence of the true signal dataset, the NRMSE is calculated between the measured dataset and each of the imputed datasets (which is the usual procedure adopted in the literature). Thus the NRMSE result is less reliable in the patient data study, which highlights further the need for evaluating imputation on the basis of other factors, such as classification performance.

C. Results

1. Results for the synthetic data

We have considered the model described in the previous section, for different combinations of parameters, which are displayed in Table VIII. In addition, since the signal

dataset is noise-free, the classification performance given by the signal dataset can serve as a benchmark. In the other direction, the benefit of an imputation algorithm is determined by how well imputation improves the classification accuracy of the measured dataset. The classification errors of the true signal dataset, measured dataset, and imputed datasets under different data distributions are shown in figures 22-27. The full set of figures is given on the companion website. It should be recognized that the figures are meant to illustrate certain effects and that other model parameters are fixed while the effects of changing a particular parameter are studied.

Table VIII. Simulation summary for the microarray data analysis pipeline

Parameters/methods	Values/descriptions
Gene block standard deviation	$\sigma_u = 0.3, 0.4, 0.5$
Gene block correlation	$\rho = 0.5, 0.7$
Gene block size	$D = 15$
Noise level	$\mu_e = 0.2, 0.3, 0.4$
MV rate	$r = 1, 5, 10, 15\%$
No. of marker genes	30
No. of total genes	500
Training sample size	60
Testing sample size	200
No. of repetitions for each model	150
Imputation algorithms	RAVG, KNN, LLS.L2, LLS.PC, LS, BPCA
Classification rules	LDA, 3NN, SVM
Feature selection methods	t-test,SFFS

a. Effect of noise level

Fig. 22 shows the impact of noise level (parameter μ_e in the data model) on imputation and classification. When noise level goes up (from left to right along the y-axis), the classification errors (along with the Bayes errors) of the measured dataset and the

imputed datasets all increase as expected; the classification errors of the signal dataset stay nearly the same and are consistently the smallest among all the datasets, since the signal dataset is noise-free. Relative to the signal dataset benchmark, the classification performances of imputed datasets deteriorate less than that of the measured dataset as the noise level increases, although their performances degrade with increasing noise. For the smallest noise level, imputation does little to improve upon the measured dataset.

b. Effect of variance

The effect of variance (parameter σ_u in the data model) on imputation and classification is shown in Fig. 23. As the variance increases, the classification errors of all datasets increase as expected. When the variance is small (e.g. $\sigma_u = 0.3$), all imputed datasets outperform the measured dataset consistently across all the combinations of feature selection methods and classification rules; however, when the variance is relatively large (e.g. $\sigma_u = 0.5$), the measured dataset catches up with and may outperform the datasets imputed by less advanced imputation methods, such as RAVG and KNNimpute. As variance increases, the discriminant power residing in the data is weakened, and the underlying data structure becomes more complex (as confirmed by computing the entropy of the eigenvalues of the covariance matrix of the gene expression matrix [92], data not shown). Thus it becomes harder for the imputation algorithms to estimate the MVs.

In addition, it is observed that the classification performance of one imputed dataset may outperform that of the other imputed dataset for a certain combination of feature-selection method and classification rule, while the performances of the two may reverse for another combination of feature selection and classification rule. For instance, when the classification rule is LDA and the feature selection method is t-

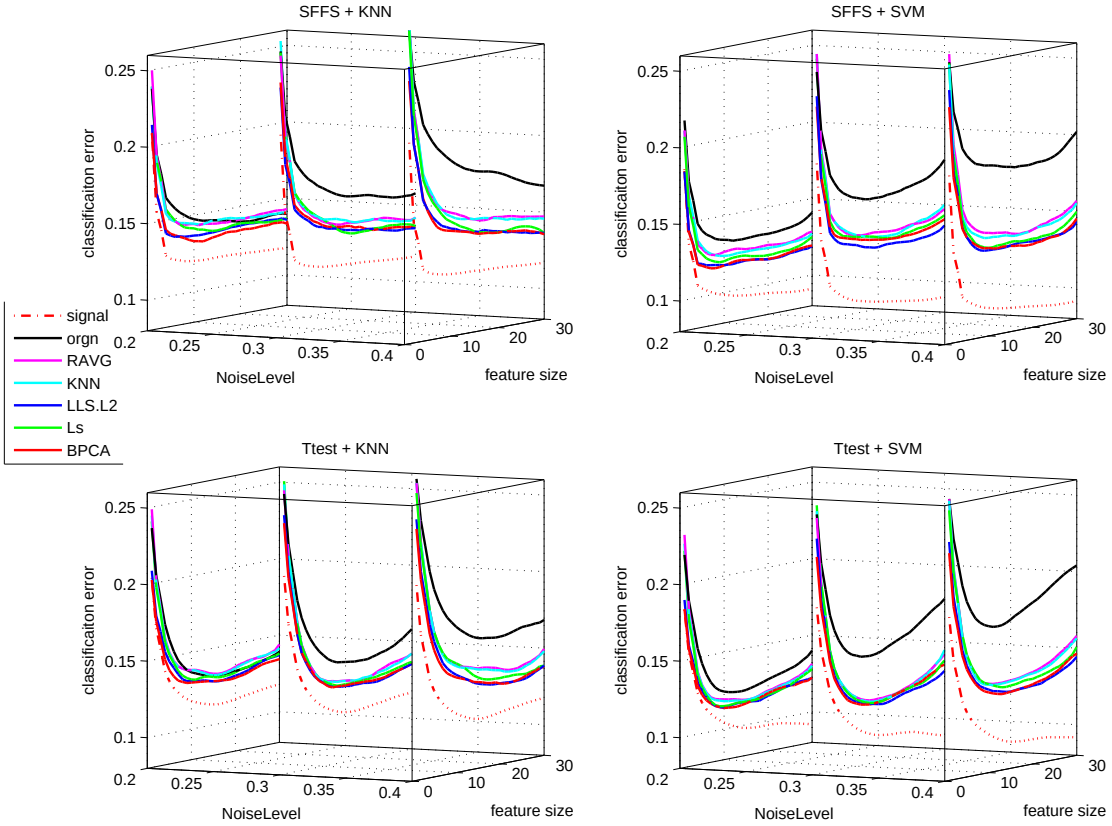


Fig. 22. Effect of noise level. The classification error of the signal dataset (signal), the measured dataset (orgn), and the five imputed datasets. The underlying distribution parameters are: SD $\sigma_u = 0.4$, gene correlation $\rho = 0.7$, MV rate $r = 10\%$. Each panel in the figure corresponds to one combination of the feature selection methods and the classification rules, which is given by the title. The x-axis labels the number of selected genes, the y-axis is the noise level, and the z-axis is the classification error.

test, the BPCA imputed dataset outperforms the LLS.L2 imputed dataset; however, the latter outperforms the former when the feature selection method is SFFS and the same classification rule is used (plots on companion website). This suggests that a certain combination of feature-selection method and classification rule may favor one imputation method over another.

c. Effect of correlation

Fig. 24 illustrates the effect of gene correlation (parameter ρ in the data model) on imputation and classification. As the gene correlation goes up, the classification errors of all datasets increase as expected. Although it is not straightforward to compare the classification performances of different datasets under different correlations, we notice that the correlation-based MV imputation methods such as LLS.PC and LS may slightly outperform BPCA in larger correlation cases, suggesting that the local correlation structure of a dataset may be better captured by such methods.

d. Effect of MV rate

Perhaps the most important observations concern the missing value rate, which is determined by adjusting the parameter τ in Eq. 5.6 to obtain a specified percentage r of missing values: $r = 1, 5, 10, 15, 20, 25\%$. Because we wish to show the effects of two model parameters, we will limit ourselves in the paper to considering 3NN and SVM with t-test feature selection. Corresponding results for other cases are on the companion website. Figures 25, 26, and 27 provide the results for the signal standard deviation $\sigma_u = 0.3, 0.4$, and 0.5, respectively, with parts a, b, and c of each figure corresponding to noise levels $\mu_e = 0.2, 0.2, 0.3, 0.3$, and $0.4, 0.4$, respectively. In all cases, $\rho = 0.7$. In Fig. 25(a), we observe the following phenomenon: there is improvement on the performance of the various imputation methods as the MV

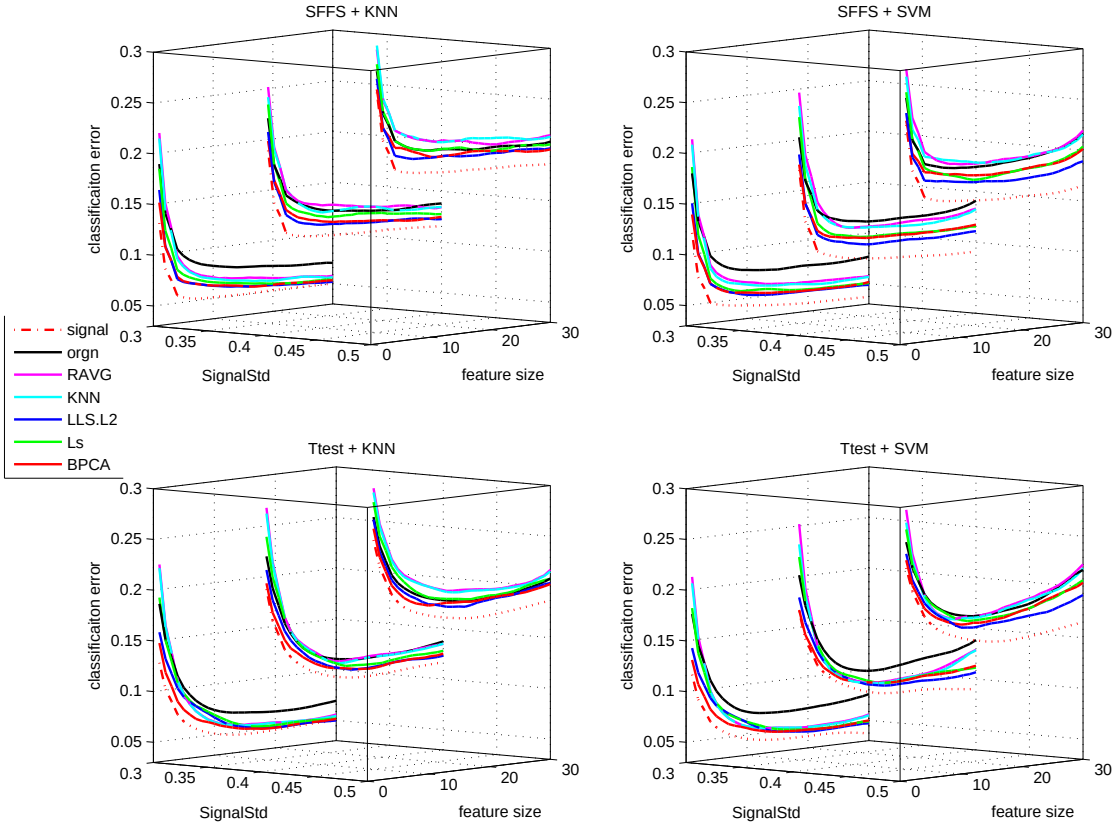


Fig. 23. Effect of variance. The classification error of the signal dataset (signal), the measured dataset (orgn), and the five imputed datasets. The underlying distribution parameters are: noise level $\mu_e = 0.2$, gene correlation $\rho = 0.7$, MV rate $r = 15\%$. Each panel in the figure corresponds to one combination of the feature selection methods and the classification rules, which is given by the title. The x-axis labels the number of selected genes, the y-axis is the signal SD, and the z-axis is the classification error.

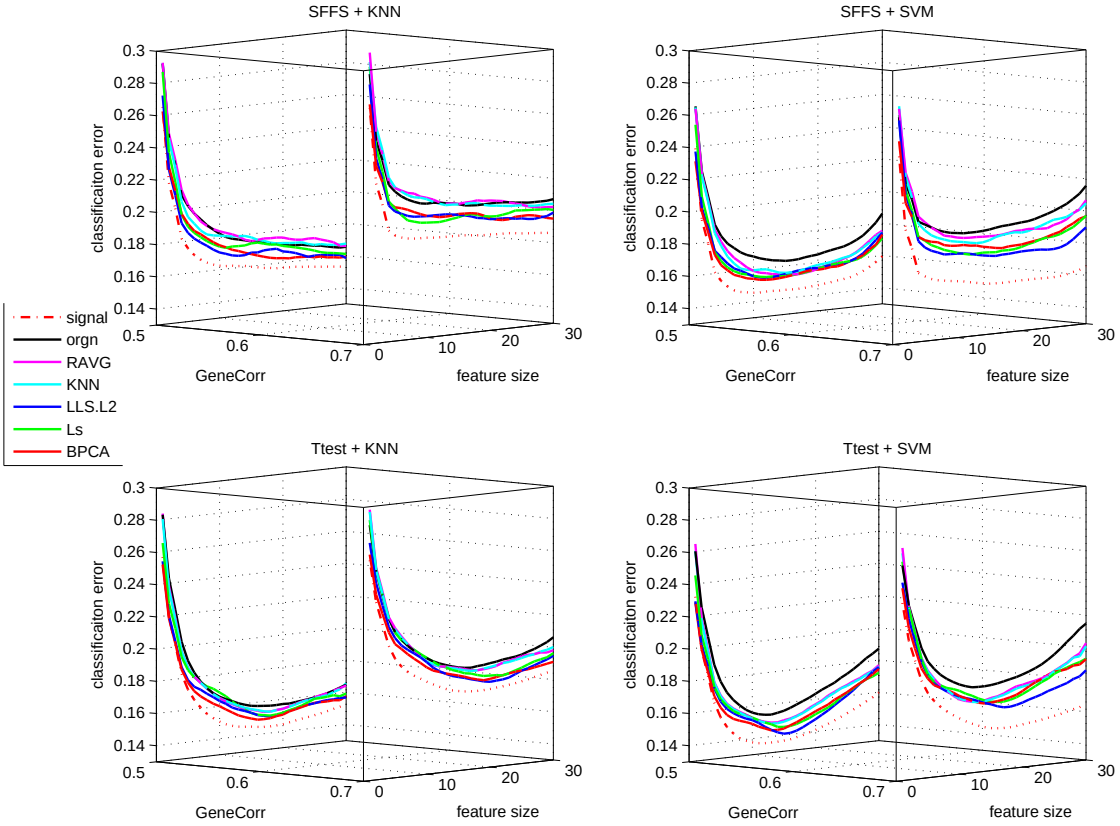


Fig. 24. Effect of correlation. The classification error of the signal dataset (signal), the measured dataset (orgn), and the five imputed datasets. The underlying distribution parameters are: SD $\sigma_u = 0.5$, noise level $\mu_e = 0.2$, MV rate $r = 10\%$. Each panel in the figure corresponds to one combination of the feature selection methods and the classification rules, which is given by the title. The x-axis labels the number of selected genes, the y-axis is the gene correlation strength, and the z-axis is the classification error.

rate initially increases, and then performance deteriorates (quickly, in some cases), as the MV rate continues to increase after a certain point. We shall refer to this phenomenon as the *missing-value rate peaking phenomenon*. It is important to stress that degradation of performance of imputation at larger MV rates is quite noticeable: at 20% the weaker imputation methods perform worse than the measured data and at 25% imputation is detrimental for kNN and not helpful for SVM. In Fig. 25(b) we again observe the MV rate peaking phenomenon; however, imputation performs better relative to the measured data. Imputation remains better throughout for SVM and only gets worse for kNN at MV rate 25%. In Fig. 25(c), the peaking phenomenon is again noticeable, but for this noise level imputation is much better relative to the measured data and all imputation methods remain better at all MV rates. Similar trends are observed in figures 26 and 27, the difference being that as σ_u increases from 0.3 to 0.4 and 0.5, the imputation methods perform increasingly worse with respect to the measured data. Note particularly the degraded performance of the simpler imputation schemes.

Fig. 28 displays the behavior of NRMSE as a function of MV rate. Here, we also observe a peaking phenomenon for the NRMSE, though a modest one. This is in contrast to previous studies, which all generally report the NRMSE to increase monotonically with increasing MV rate [86, 87, 91, 95]; this may be a consequence of the different way in which the MVs are selected in those studies as compared with the present one; in the former, MVs are picked randomly, whereas in the latter, MVs are picked based on quality considerations, revealing the peaking phenomenon.

2. Results for the patient data

For the patient data, since the true signal is unknown, we only conduct the comparison of imputations with respect to different MV rates. The effect of MV rate is shown in

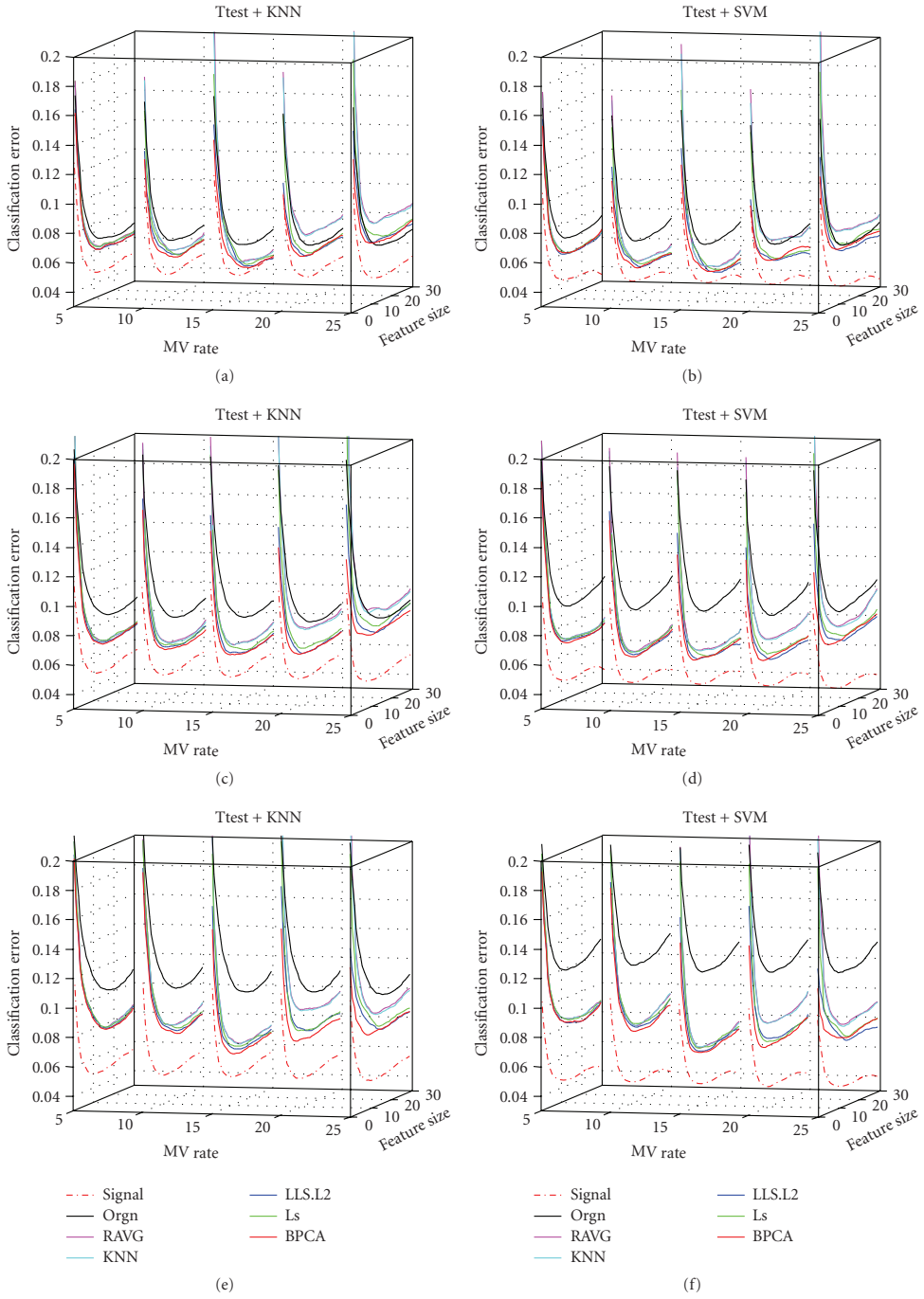


Fig. 25. Effect of MV Rate. The classification error of the signal dataset (signal), the measured dataset (orgn), and the five imputed datasets. The underlying distribution parameters are: SD $\sigma_u = 0.3$, gene correlation $\rho = 0.7$ and noise level $\mu_e = 0.2, 0.2, 0.3, 0.3, 0.4, 0.4$ for subfigures (a)-(f), respectively. The x-axis labels the number of selected genes, the y-axis is the MV rate, and the z-axis is the classification error.

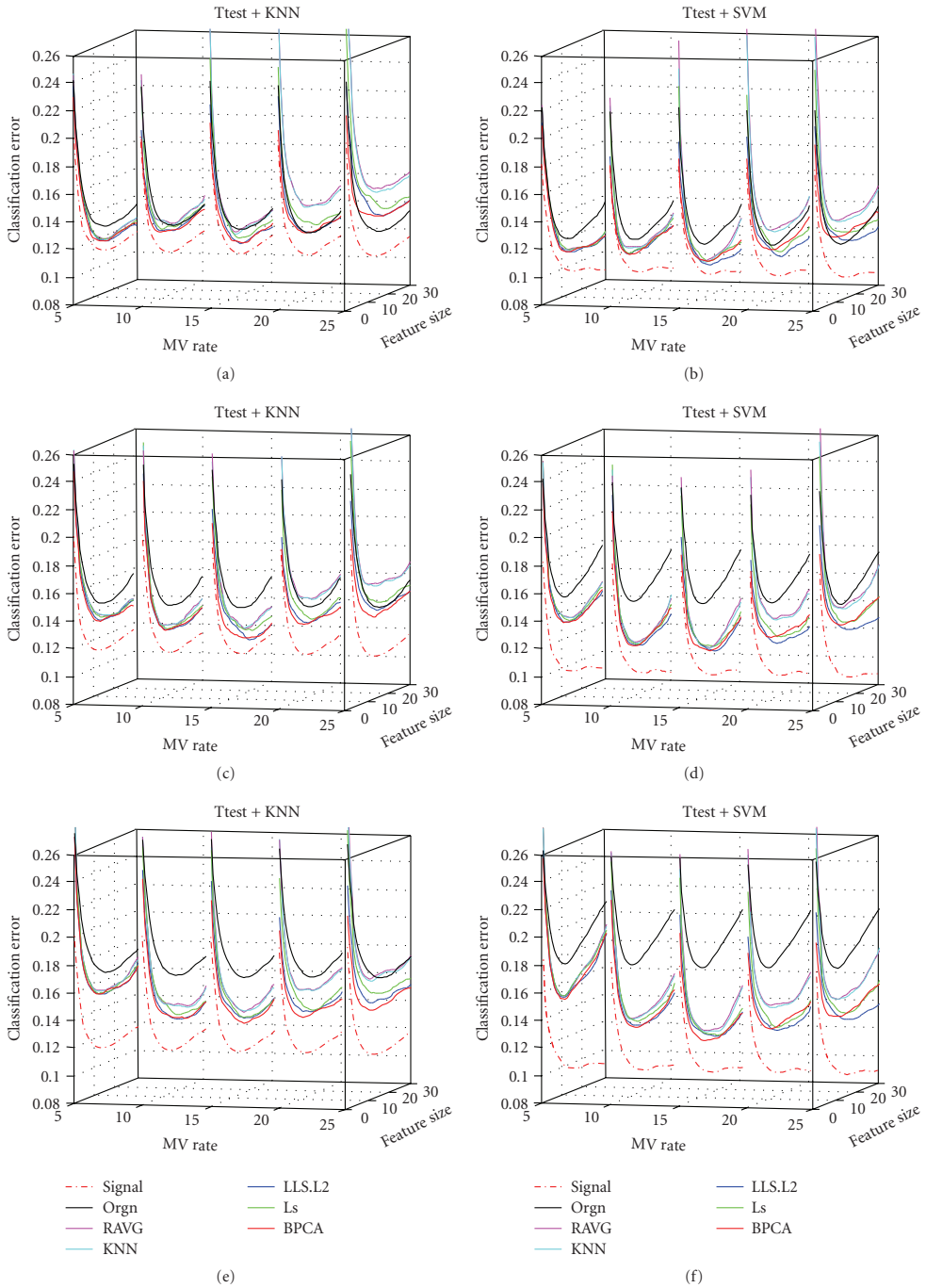


Fig. 26. Effect of MV Rate. The classification error of the signal dataset (signal), the measured dataset (orgn), and the five imputed datasets. The underlying distribution parameters are: SD $\sigma_u = 0.4$, gene correlation $\rho = 0.7$ and noise level $\mu_e = 0.2, 0.2, 0.3, 0.3, 0.4, 0.4$ for subfigures (a)-(f), respectively. The x-axis labels the number of selected genes, the y-axis is the MV rate, and the z-axis is the classification error.

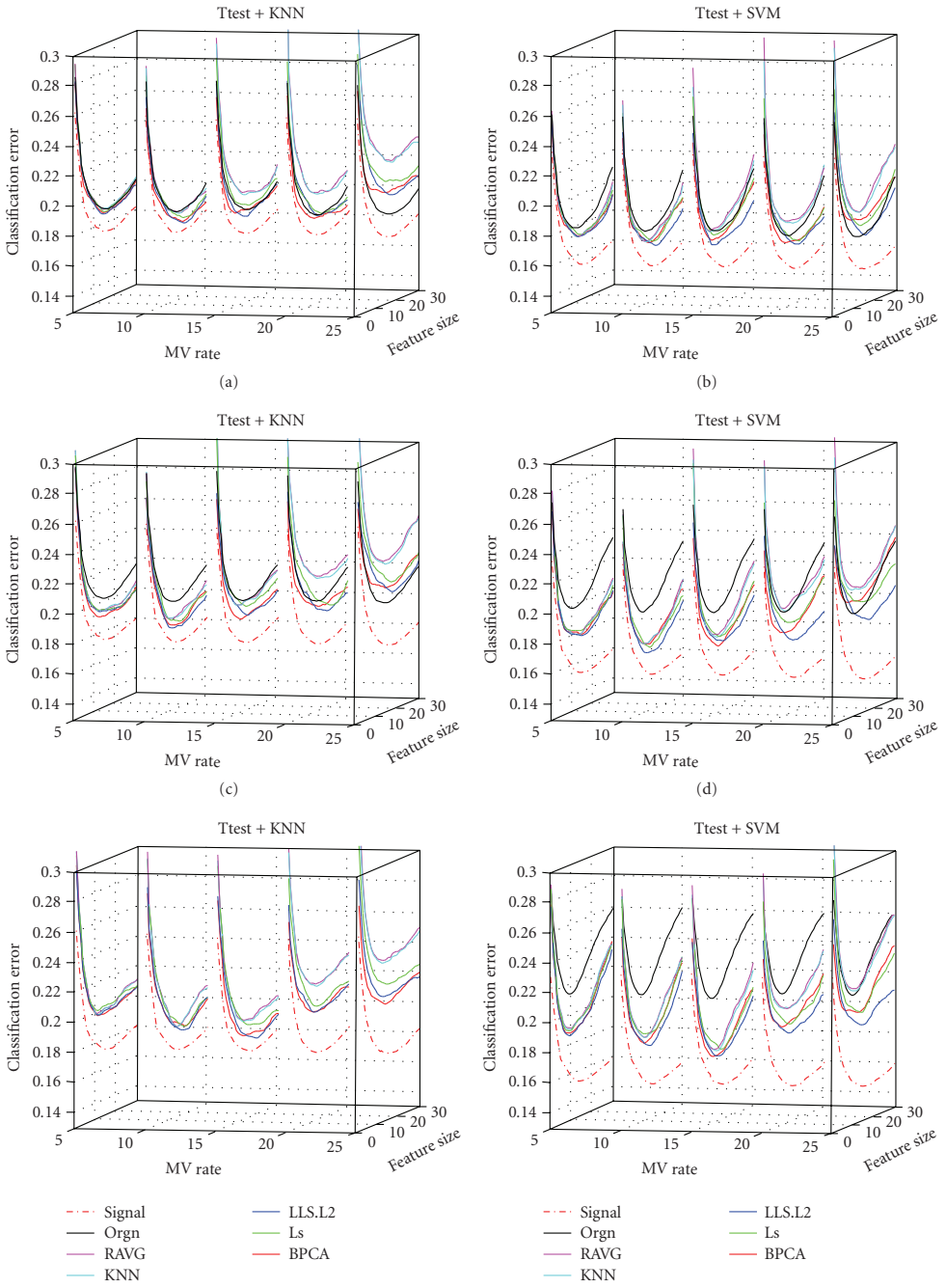


Fig. 27. Effect of MV Rate. The classification error of the signal dataset (signal), the measured dataset (orgn), and the five imputed datasets. The underlying distribution parameters are: SD $\sigma_u = 0.5$, gene correlation $\rho = 0.7$ and noise level $\mu_e = 0.2, 0.2, 0.3, 0.3, 0.4, 0.4$ for subfigures (a)-(f), respectively. The x-axis labels the number of selected genes, the y-axis is the MV rate, and the z-axis is the classification error.

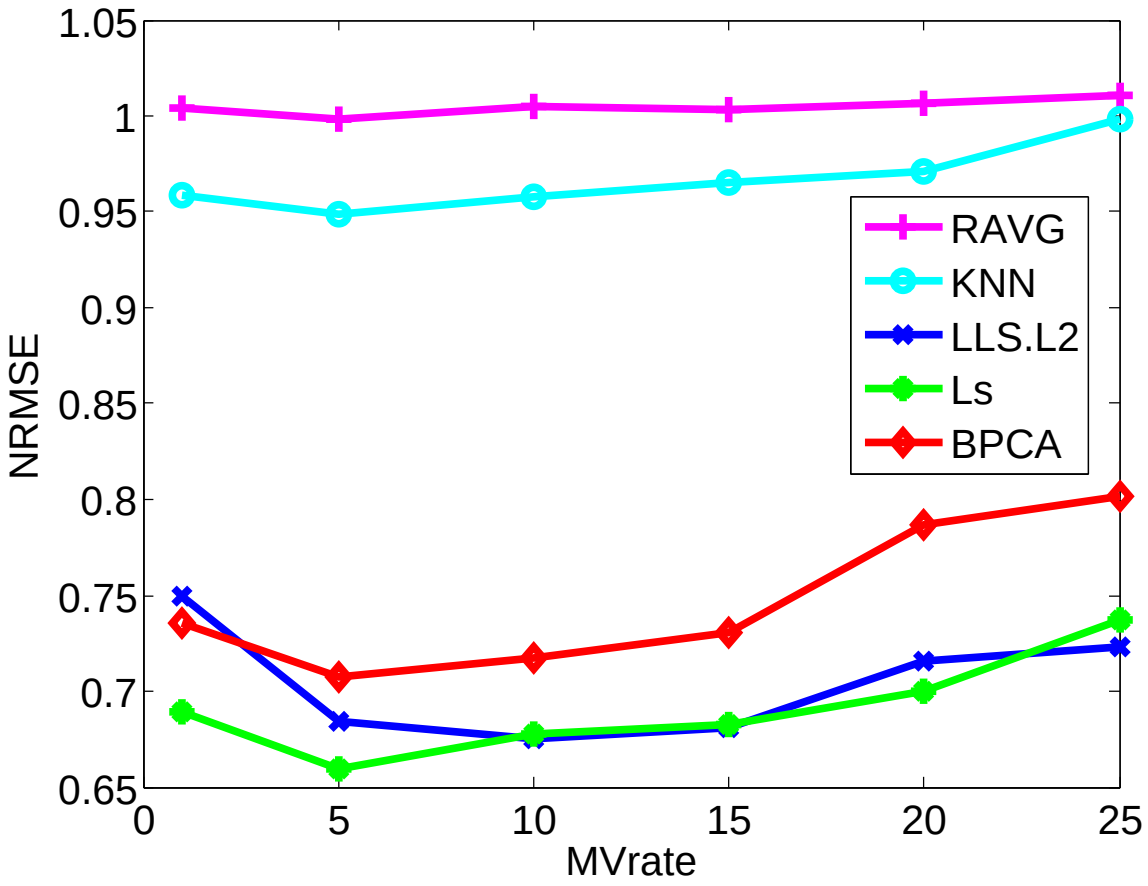


Fig. 28. The NRMSE values (y-axis) of the five imputation algorithms with respect to the MV rate (x-axis). The underlying distribution parameters are: SD $\sigma_u = 0.5$, noise level $\mu_e = 0.2$, gene correlation $\rho = 0.7$.

figures 29 and 30, for the BREAST and the PROST dataset, respectively. The trends observed are similar to those in the synthetic data study, in the sense that there is a degradation of performance of imputation methods with increasing MV rates. On the other hand, the missing-value rate peaking phenomenon is less evident here, but still present, as can be seen from the fact that the classification performance of LLS, LS and BPCA imputed datasets in a few cases becomes better under a larger MV rate than the corresponding datasets with a smaller MV rate.

It is again observed that the classification performances of imputed datasets depend on the underlying combination of feature selection method and classification rule. For example, RAVG and KNNimpute show satisfactory performances for the combinations SFFS+LDA and Ttest+LDA (data not shown), but perform relatively poorly for the other combinations.

The NRMSE values of different imputation methods generally decrease first and then increase as the MV rate increases (see Fig. 31) which is similar to the trend observed in synthetic data study.

It is also found that there is no strong correlation between the low-level performance measure NRMSE and the high-level measure classification error. A small NRMSE may not necessarily suggest a small classification error, i.e. an imputation method may perform better than another imputation method in terms of estimation accuracy, but the former may not be as good as the latter in terms of classification performance. In other words, although a given imputation method may be more accurate than another when measured by NRMSE, it might decrease more the discrimination power presents in the original data.

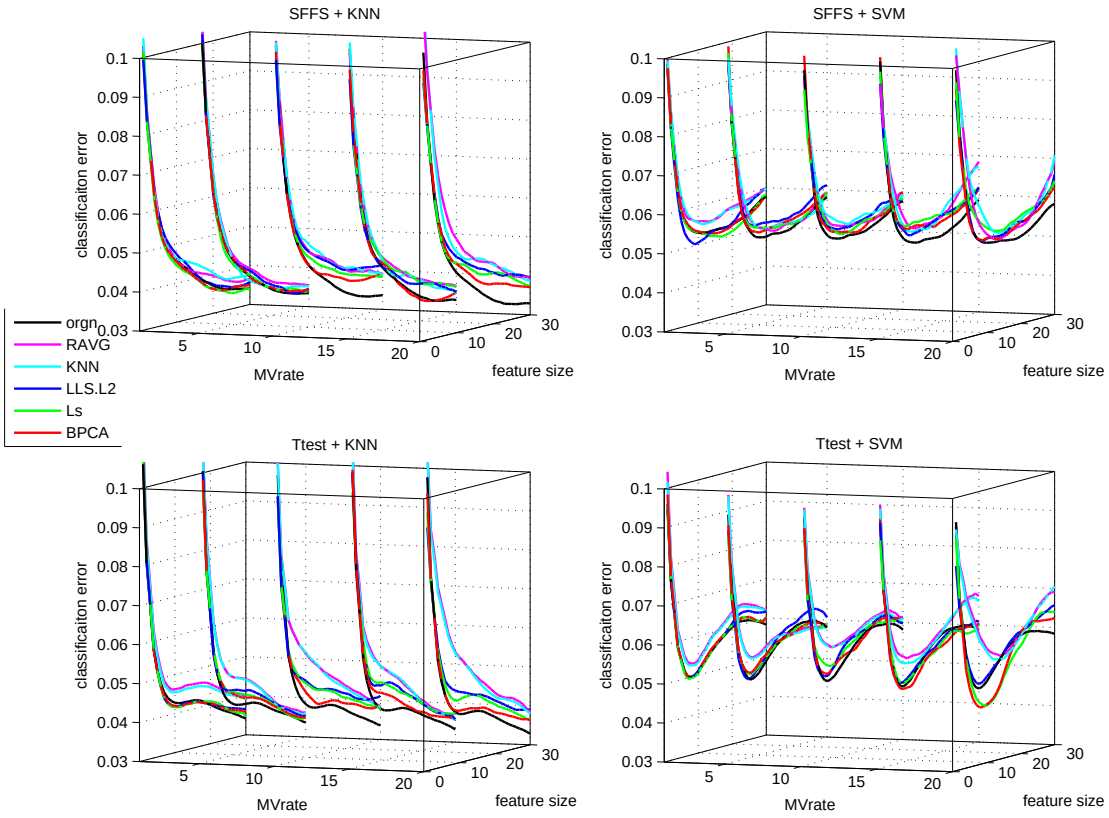


Fig. 29. The classification errors of the measured prostate cancer dataset (orgn), and the five imputed datasets. Each panel in the figure corresponds to one combination of the feature selection methods and the classification rules, which is given by the title. The x-axis labels the number of selected genes, the y-axis is the MV rate, and the z-axis is the classification error.

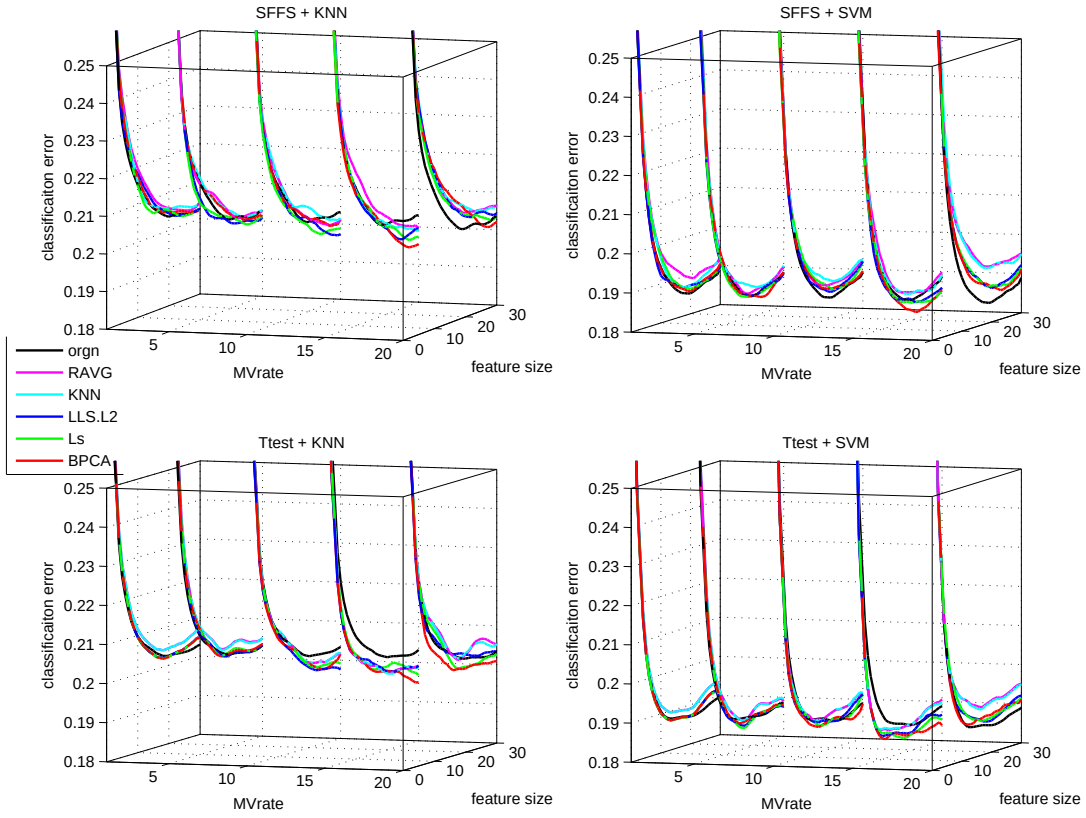
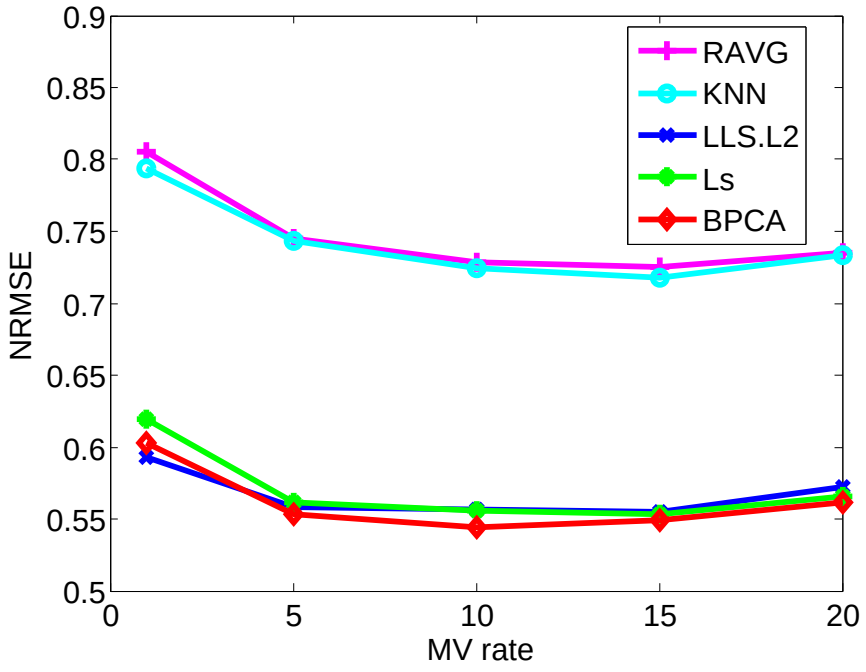
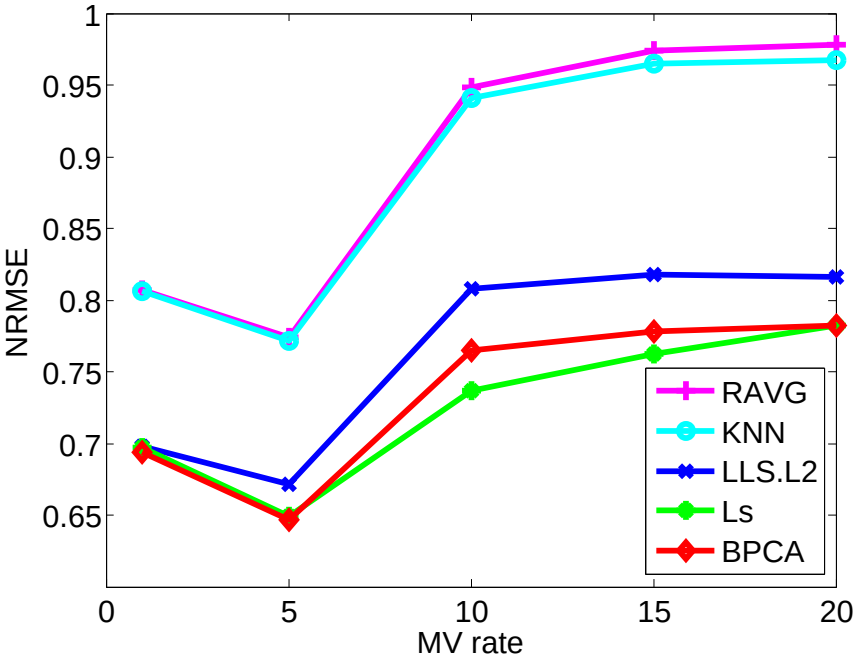


Fig. 30. The classification errors of the measured breast cancer dataset (orgn), and the five imputed datasets. The meanings of the axes and titles are the same as in the previous figure.



(a)



(b)

Fig. 31. The NRMSE values (y-axis) of the five imputation algorithms with respect to the MV rate (x-axis) for the PROST dataset and the BREAST dataset.

REFERENCES

[1] S. H. Woolf, "The meaning of translational research and why it matters," *J of Am Med Assoc*, vol. 299, pp. 211–213, 2008.

[2] E. R. Dougherty and M. L. Bittner, *Epistemology of the Cell: A Systems Perspective on Biological Knowledge*, Wiley-IEEE Press, New Jersey, 2011.

[3] M. Kerr, M. Martin, and G. Churchill, "Analysis of variance for gene expression microarray data," *Computational Biology*, vol. 7, pp. 819–837, 2000.

[4] J. Peng, J. E. Elias, C. C. Thoreen, L. J. Licklider, and S. P. Gygi, "Evaluation of multidimensional chromatography coupled with tandem mass spectrometry (LC/LC-MS/MS) for large-scale protein analysis: the yeast proteome," *Journal of Proteome Research*, vol. 2, pp. 43–50, 2003.

[5] S. P. Gygi and J. E. Elias, "Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry," *Nature Methods*, vol. 4, no. 3, pp. 207–214, 2007.

[6] N. Rifai, M. A. Gillette, and S. A. Carr, "Protein biomarker discovery and validation: the long and uncertain path to clinical utility," *Nature Biotechnology*, vol. 24, pp. 971–983, 2006.

[7] A. Pandey, J. S. Andersen, and M. Mann, "Use of mass spectrometry to study signaling pathways," *Science's STKE*, vol. 37, pp. pl1, 2000.

[8] J. A. Hewel, J. Liu, K. Onishi, V. Fong, and et al., "Synthetic peptide arrays for pathway-level protein monitoring by LC-MS/MS," *Mol Cell Proteomics*, vol. 9, pp. 2460–2473, 2010.

[9] R. Frank and R. Hargreaves, "Clinical biomarkers in drug discovery and development," *Nat Rev Drug Disc*, vol. 2, pp. 566–580, 2003.

[10] C. Hop and R. Bakhtiar, "An introduction to electrospray ionization and matrix-assisted laser desorption/ionization mass spectrometry: essential tools in a modern biotechnology environment," *Biospectroscopy*, vol. 3, pp. 259–28, 1997.

[11] M. Karas and U. Bahr, "Laser desorption ionization mass spectrometry of large biomolecules," *Trends Anal Chem*, vol. 9, pp. 321–325, 1990.

[12] S. Batoy, E. Akhmetova, S. Miladinovic, J. Smeal, and C. L. Wilkins, "Developments in MALDI mass spectrometry: the quest for the perfect matrix," *Appl Spectrosc Rev*, vol. 43, pp. 485–550, 2008.

[13] Q. Hu, R. J. Noll, H. Li, A. Makarov, M. Hardman, and R. Graham Cooks, "The Orbitrap: a new mass spectrometer," *J Mass Spectrom*, vol. 40, pp. 430–443, 2005.

[14] J. F. J. Todd and R. E. March, *Quadrupole Ion Trap Mass Spectrometry*, Wiley-Interscience, New York, 2005.

[15] H. Wollnik, "Time-of-flight mass analyzers," *Mass Spectrometry Reviews*, vol. 12, pp. 89–11, 1993.

[16] I. V. Chernushevich, A. V. Loboda, and B. A. Thomson, "An introduction to Quadrupole-time-of-flight mass spectrometry," *J Mass Spectrom*, vol. 36, pp. 849–865, 2001.

[17] M. L. Gross and D. L. Rempel, "Fourier transform mass spectrometry," *Science*, vol. 226, pp. 261C268, 1984.

[18] X. Li, E. C. Yi, C. J. Kemp, H. Zhang, and R. Aebersold, "A software suite for the generation and comparison of peptide arrays from sets of data collected by liquid chromatography-mass spectrometry," *Mol Cell Proteomics*, vol. 4, pp. 1328–40, 2005.

[19] O. Schulz-Trieglaff, N. Pfeifer, C. Gröpl, O. Kohlbacher, and K. Reinert, "LC-MSsim – a simulation software for liquid chromatography mass spectrometry data," *BMC Bioinformatics*, vol. 9, pp. 423, 2008.

[20] D. N. Perkins, D. J. Pappin, D. M. Creasy, and J. S. Cottrell, "Probability based protein identification by searching sequence databases using mass spectrometry data," *Electrophoresis*, vol. 20, pp. 3551–67, 1999.

[21] A. I. Nesvizhskii, F. F. Roos, J. Grossmann, M. Vogelzang, J. S. Eddes, W. Gruissem, S. Baginsky, and R. Aebersold, "Dynamic spectrum quality assessment and iterative computational analysis of shotgun proteomic data: toward more efficient idenfitication of post-translational modifications, sequence polymorphisms, and novel peptides," *Mol Cell Proteomics*, vol. 5, pp. 652–670, 2006.

[22] M. Bantscheff, M. Schirle, G. Sweetman, J. Rick, and B. Kuster, "Quantitative mass spectrometry in proteomics: a critical review," *Anal Bioanal Chem*, vol. 389, pp. 1017–1031, 2007.

[23] B. Domon and R. Aebersold, "Mass spectrometry and protein analysis," *Science*, vol. 312, pp. 212–7, 2006.

[24] J. S. Morris, K. R. Coombes, J. Koomen, K. A. Baggerly, and R. Kobayashi, "Feature extraction and quantification for mass spectrometry in biomedical

applications using the mean spectrum," *Bioinformatics*, vol. 21, pp. 1764–1775, 2005.

[25] K. R. Coombes, S. Tsavachidis, J. S. Morris, K. A. Baggerly, M. C. Hung, and H. M. Kuerer, "Improved peak detection and quantification of mass spectrometry data acquired from surface-enhanced laser desorption and ionization by denoising spectra with the undecimated discrete wavelet transform," *Proteomics*, vol. 5, pp. 4107–4117, 2005.

[26] P. Du, W. A. Kibbe, and S. M. Lin, "Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching," *Bioinformatics*, vol. 22, pp. 2059–2065, 2006.

[27] Y. Wang, X. Zhou, H. Wang, K. Li, L. Yao, and S. T. C. Wong, "Reversible jump MCMC approach for peak identification for stroke SELDI mass spectrometry using mixture model," *Bioinformatics*, vol. 24, pp. i407–i413, 2008.

[28] B. Y. Renard, M. Kirchner, J. A. Steen, and F. A. Hamprecht, "NITPICK: peak identifcation for mass spectrometry data," *BMC Bioinformatics*, vol. 9, pp. 355–370, 2008.

[29] X. Li, E. C. Yi, C. J. Kemp, H. Zhang, and R. Aebersold, "A software suite for the generation and comparison of peptide arrays from sets of data collected by liquid chromatography-mass spectrometry," *S Mol Cell Proteom*, vol. 4, pp. 1328–1340, 2005.

[30] N. Jaitly, A. Mayampurath, K. Littlefield, J. N. Adkins, G. A. Anderson, and R. D. Smith, "Decon2LS: An open-source software package for automated processing and visualization of high resolution mass spectrometry data," *BMC Bioinformatics*, vol. 10, pp. 87–101, 2009.

[31] M. R. Hoopmann, G. L. Finney, and M. J. MacCoss, "High speed data reduction, feature selection, and MS/MS spectrum quality assessment of shotgun proteomics datasets using high resolution mass spectrometry," *Anal Chem*, vol. 79, pp. 5630–5632, 2007.

[32] M. Katajamaa and M. Oresic, "Processing methods for differential analysis of lc/ms profile data," *BMC Bioinformatics*, vol. 6, pp. 179–190, 2005.

[33] M. Bellew, M. Coram, M. Fitzgibbon, M. Igra, T. Randolph, P. Wang, and et al., "A suite of algorithms for the comprehensive analysis of complex protein mixtures using high-resolution LC-MS," *Bioinformatics*, vol. 22, pp. 1902–1909, 2006.

[34] L. Mueller, O. Rinner, A. Schmidt, S. Letarte, B. Bodenmiller, M. Brusniak, O. Vitek, R. Aebersold, and M. Muller, "SuperHirn–a novel tool for high resolution LC-MS-based peptide/protein profiling," *Proteomics*, vol. 7, pp. 3470–80, 2007.

[35] M. E. Monroe, N. Tolic, N. Jaitly, J. L. Shaw, J. N. Adkins, and R. D. Smith, "VIPER: an advanced software package to support high-throughput LC-MS peptide identification," *Bioinformatics*, vol. 23, pp. 2021–2023, 2007.

[36] J. Cox and M. Mann, "MaxQuant enables high peptide identification rates, individualized p.p.b-range mass accuracies and proteome-wide protein quantification," *Nature Biotechnology*, vol. 26, pp. 1367–372, 2008.

[37] M. Sturm, A. Bertsch, C. Gröpl, A. Hildebrandt, R. Hussong, and et al., "OpenMS – an open-source software framework for mass spectrometry," *BMC Bioinformatics*, vol. 9, pp. 163–173, 2008.

[38] A. Brevern, S. Hazout, and A. Malpertuy, "Influence of microarrays experiments missing values on the stability of gene groups by hierarchical clustering," *Bioinformatics*, vol. 5, pp. 114–124, 2004.

[39] Y. Sun, J. Zhang, U. M. Braga-Neto, and E. R. Dougherty, "BPDA – a Bayesian peptide detection algorithm for mass spectrometry," *BMC Bioinformatics*, vol. 11, pp. 490–500, 2010.

[40] Y. Sun, J. Zhang, U. M. Braga-Neto, and E. R. Dougherty, "BPDA2d – a 2D global optimization based Bayesian peptide detection algorithm for LC-MS," *Bioinformatics*, vol. 28, pp. 564–572, 2012.

[41] Y. Sun, U. M. Braga-Neto, and E. R. Dougherty, "Modeling and systematic analysis of the LC-MS proteomics pipeline," *Submitted to BMC Genomics*, 2012.

[42] Y. Sun, U. M. Braga-Neto, and E. R. Dougherty, "Impact of missing value imputation on classification for DNA microarray gene-expression data: a model-based study," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 50, pp. 17 pages, 2009.

[43] K. Noy and D. Fasulo, "Improved model-based, platform-independent feature extraction for mass spectrometry," *Bioinformatics*, vol. 23, pp. 2528–2535, 2007.

[44] A. L. Rockwood, S. L. Van Orden, and R. D. Smith, "Rapid cacluation of isotope distributions," *Anal Chem*, vol. 67, pp. 2699–2704, 1995.

[45] M. W. Senko, S. C. Beu, and F. W. McLafferty, "Determination of monoisotopic masses and ion populations for large biomolecules from resolved isotopic distributions," *J Am Soc Mass Spectrom*, vol. 6, pp. 229–233, 1995.

[46] P. Du, R. Sudha, M. B. Prystowsky, and R. H. Angeletti, "Data reduction of isotope-resolved LC-MS spectra," *Bioinformatics*, vol. 23, pp. 1394–1400, 2007.

[47] P. Du and R. H. Angeletti, "Automatic deconvolution of isotope-resolved mass spectra using variable selection and quantized peptide mass distribution," *Anal Chem*, vol. 78, pp. 3385–3392, 2006.

[48] P. Du, G. Stolovitzky, P. Horvatovich, R. Bischoff, J. Lim, and F. Suits, "A noise model for mass spectrometry based proteomics," *Bioinformatics*, vol. 24, no. 8, pp. 1070–1077, 2008.

[49] K. R. Coombes, J. Koomen, K. A. Baggerly, J. S. Morris, and R. Kobayashi, "Understanding the characteristics of mass spectrometry data through the use of simulation," *Cancer Informatics*, vol. 1, no. 1, pp. 41–52, 2005.

[50] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans Pattern Anal Mach Intell*, vol. 6, pp. 721–741, 1984.

[51] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, 2004.

[52] P. Kaur and P. B. O'Connor, "Use of statistical methods for estimation of total number of charges in a mass spectrometry experiment," *Anal Chem*, vol. 76, pp. 2756–2762, 2004.

[53] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1979.

[54] G. Burgers, P. J. Leeuwen, and G. Evensen, "Analysis scheme in the ensemble Kalman filter," *Monthly Weather Review*, vol. 126, pp. 1719–1724, 1998.

[55] R. Duda and P. Hart, *Pattern Classification*, JohnWiley&Sons, New York, NY, USA, 2001.

[56] D. M. Horn, R. A. Zubarev, and F. W. McLafferty, "Automated reduction and interpretation of high resolution electrospray mass spectra of large molecules," *J Am Soc Mass Spectrom*, vol. 11, no. 4, pp. 320–332, 2000.

[57] D. A. Stead, A. Preece, and J. P. Brown, "Universal metrics for quality assessment of protein identifications by mass spectrometry," *Mol Cell Proteomics*, vol. 5, pp. 1205–1211, 2006.

[58] L. McHugh and J. W. Arthur, "Computational methods for protein identification from mass spectrometry data," *PLoS Comput Biol*, vol. 4, pp. e12–23, 2008.

[59] M. Dijkstra and R. C. Jansen, "Optimal analysis of complex protein mass spectra," *Proteomics*, vol. 9, pp. 3869–3876, 2009.

[60] K. C. Leptos, D. A. Sarracino, J. D. Jaffe, B. Krastins, and G. M. Church, "MapQuant: Open-source software for large-scale protein quantification," *Proteomics*, vol. 6, pp. 1770–1782, 2006.

[61] G. Schwarz, "Estimating the dimension of a model," *Ann Stat*, vol. 6, pp. 461–464, 1978.

[62] V. B. Di Marco and G. G. Bombi, "Mathematical functions for the representation of chromatographic peaks," *Journal of Chromatography A*, vol. 931, pp. 1–30, 2001.

[63] J. Zhang, E. Gonzalez, T. Hestilow, W. Haskins, and Y. Huang, "Review of peak detection algorithms in liquid-chromatography-mass spectrometry," *Curr*

Genomics, vol. 10, pp. 388–401, 2009.

[64] J. Klimek, J. Eddes, L. Hohmann, J. Jackson, A. Peterson, and et al., "The standard protein mix database: A diverse dataset to assist in the production of improved peptide and protein identification software tools," *Journal of Proteome Research*, vol. 7, pp. 96–103, 2008.

[65] W. E. Haskins, K. Petritis, and J. Zhang, "MRCQuant – an accurate LC-MS relative isotopic quantification algorithm on TOF instruments," *BMC Bioinformatics*, vol. 12, pp. 74–85, 2011.

[66] C. Knox, V. Law, T. Jewison, P. Liu, S. Ly, and et al., "Drugbank 3.0: a comprehensive resource for 'omics' research on drugs," *Nucleic Acids Research*, vol. 39, pp. D1035–41, 2011.

[67] Y. Taniguchi, P. J. Choi, G. Li, H. Chen, M. Babu, and et al., "Quantifying E. coli proteome and transcriptome with single-molecule sensitivity in single cells," *Science*, vol. 329, pp. 533–538, 2010.

[68] P. Lu, C. Vogel, R. Wang, X. Yao, and E. M. Marcotte, "Absolute protein expression profiling estimates the relative contributions of transcriptional and translational regulation," *Nature Biotechnology*, vol. 25, pp. 117–24, 2007.

[69] J. Hua, T. Waibhav, and E. R. Dougherty, "Performance of feature selection methods in the classification of high-dimensional data," *Pattern Recognition*, vol. 42, pp. 409–424, 2008.

[70] W. Timm, A. Scherbart, S. Bocker, O. Kohlbacher, and T. W. Nattkemper, "Peak intensity prediction in MALDI-TOF mass spectrometry: A machine

learning study to support quantitative proteomics," *BMC Bioinformatics*, vol. 9, pp. 443–460, 2008.

[71] N. B. Cech and C. G. Enke, "Practical implications of some recent studies in electrospray ionization fundamentals," *Mass Spectrom Rev*, vol. 20, no. 6, pp. 362–87, 2001.

[72] M. Anderle, S. Roy, H. Lin, C. Becker, and K. Joho, "Quantifying reproducibility for differential proteomics: noise analysis for protein liquid chromatography-mass spectrometry of human serum," *Bioinformatics*, vol. 20, no. 18, pp. 3575–3582, 2004.

[73] A. T. Iavarone, J. C. Jurchen, and E. R. Williams, "Effects of solvent on the maximum charge state and charge state distribution of protein ions produced by electrospray ionization," *J Am Soc Mass Spectrom*, vol. 11, no. 11, pp. 976–985, 2000.

[74] L. Konermann, "A minimalist model for exploring conformational effects on the electrospray charge state distribution of proteins," *J Phys Chem B*, vol. 111, pp. 6534–6543, 2007.

[75] J. Zhang and W. Haskins, "ICPD- a new peak detection algorithm for LC/MS," *BMC Genomics*, vol. 11, pp. S8–18, 2010.

[76] J. R. Yates, J. K. Eng, A. L. McCormack, and D. Schieltz, "Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database," *Anal Chem*, vol. 67, pp. 1426–1436, 1995.

[77] P. Mallick, M. Schirle, S. S. Chen, M. R. Flory, H. Lee, D. Martin, J. Ranish, B. Raught, R. Schmitt, T. Werner, B. Kuster, and R. Aebersold, "Computa-

tional prediction of proteotypic peptides for quantitative proteomics," *Nature Biotechnology*, vol. 25, pp. 125–131, 2007.

[78] J. R. Whiteaker, H. Zhang, J. K. Eng, and et al., "Head-to-head comparison of serum fractionation techniques," *Journal of Proteome Research*, vol. 6, no. 2, pp. 828–36, 2007.

[79] B. C. Bohrer, Y. F. Li, J. P. Reilly, D. E. Clemmer, R. D. DiMarchi, P. Radivojac, H. Tang, and R.J. Arnold, "Combinatorial libraries of synthetic peptides as a model for shotgun proteomics," *Anal Chem*, vol. 82, no. 15, pp. 6559–568, 2010.

[80] L. A. Echan, H. Y. Tang, A. K. Nadeem, K. Lee, and D. W. Speicher, "Depletion of multiple high-abundance proteins improves protein profiling capacities of human serum and plasma," *Proteomics*, vol. 5, no. 13, pp. 3292–3303, 2005.

[81] B. H. Bazzi, "Ionization competitors extend the linear range of electrospray ionization mass spectrometry," M.S. thesis, The University of Texas at Arlington, Arlington, 2010.

[82] O. Rinner, L. N. Mueller, M. Hubálek, M. Müller, M. Gstaiger, and R. Aebersold, "An integrated mass spectrometric and computational framework for the analysis of protein interaction networks," *Nature Biotechnology*, vol. 25, pp. 345–352, 2007.

[83] R.D. Smith, G. A. Anderson, M. S. Lipton, and et al., "An accurate mass tag strategy for quantitative and highthroughput proteome measurements," *Proteomics*, vol. 2, pp. 513–523, 2002.

[84] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani,

D. Botstein, and R. Altman, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, pp. 520–525, 2001.

[85] S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara, and S. Ishii, "A Bayesian missing value estimation method for gene expression profile data," *Bioinformatics*, vol. 19, pp. 2088–2096, 2003.

[86] T. Bϕ, B. Dysvik, and I. Jonassen, "LSimpute: accurate estimation of missing values in microarray data with least squares methods," *Nucleic Acids Research*, vol. 32, pp. e34–41, 2004.

[87] H. Kim, G. Golub, and H. Park, "Missing value estimation for DNA microarray gene expression data: local least squares imputation," *Bioinformatics*, vol. 21, pp. 187–198, 2005.

[88] R. Jornsten, H. Wang, W. Welsh, and M. Ouyang, "DNA microarray data imputation and significance analysis of differential expression," *Bioinformatics*, vol. 21, pp. 4155–4161, 2005.

[89] M. Scholz, F. Kaplan, C. Guy, J. Kopka, and J. Selbig, "Non-linear PCA: a missing data approach," *Bioinformatics*, vol. 21, pp. 3887–3895, 2005.

[90] J. Tuikkala, L. Elo, O. Nevalainen, and T. Aittokallio, "Improving missing value estimation in microarray data with gene ontology," *Bioinformatics*, vol. 22, pp. 566–572, 2006.

[91] J. Hu, H. Li, M. Waterman, and X. Zhou, "Integrative missing value estimation for microarray data," *Bioinformatics*, vol. 7, pp. 449–462, 2006.

[92] G. Brock, J. Shaffer, R. Blakesley, M. Lotz, and G. Tseng, "Which missing value imputation method to use in expression profiles: a comparative study

and two selection schemes," *BMC Bioinformatics*, vol. 9, pp. 12–23, 2008.

[93] M. Sehgal, I. Gondal, L. Dooley, and R. Coppel, "How to improve postgenomic knowledge discovery using imputation," *EURASIP J on Bioinformatics and System Biology*, vol. 2009, pp. 14 pages, 2009.

[94] I. Scheel, M. Aldrin, I. Glad, R. Sorum, H. Lyng, and A. Frigessi, "The influence of missing value imputation on detection of differentially expressed genes from microarray data," *Bioinformatics*, vol. 21, pp. 4272–4279, 2005.

[95] J. Tuikkala, L. Elo, O. Nevalainen, and T. Aittokallio, "Missing value imputation improves clustering and interpretation of gene expression microarray data," *Bioinformatics*, vol. 9, pp. 202–215, 2008.

[96] D. Wang, Y. Lv, Z. Guo, X. Li, Y. Li, J. Zhu, D. Yang, J. Xu, C. Wang, S. Rao, and B. Yang, "Effects of replacing the unreliable cDNA microarray measurements on the disease classification based on gene expression profiles and functional modules," *Bioinformatics*, vol. 22, pp. 2883–2889, 2006.

[97] Y. Shi, Z. Cai, and G. Lin, "Classification accuracy based microarray missing value imputation," in *Bioinformatics Algorithms: Techniques and Applications*, I. Mandoiu and A. Zelikovsky, Eds., pp. 303–328. Wiley-Interscience, New Jersey, 2007.

[98] D. Hoyle, M. Rattray, R. Jupp, and A. Brass, "Making sense of microarray data distributions," *Bioinformatics*, vol. 18, pp. 576–584, 2002.

[99] R. Autio, S. Kilpinen, M. Saarela, O. Kallioniemi, S. Hautaniemi, and J. Astola, "Comparison of affymetrix data normalization methods using 6,926 experiments

across five array generations," *BMC Bioinformatics*, vol. Vol. 10 Suppl 1, pp. 1–12, 2009.

[100] M. Kerr, M. Martin, and G. Churchill, "Statistical design and the analysis of gene expression microarray data," *Genetic Research*, vol. 77, pp. 123–128, 2000.

[101] S. Attoor, E. R. Dougherty, Y. Chen, M. Bittner, and J. Trent, "Which is better for cDNA-microarray-based classification: ratios or direct intensities," *Bioinformatics*, vol. 20, pp. 2513–2520, 2004.

[102] G. Tseng, M. Oh, L. Rohlin, J. Liao, and W. Wong, "Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects," *Nucleic Acids Research*, vol. 29, pp. 2549–2557, 2001.

[103] I. Shmulevich and W. Zhang, "Binary analysis and optimization-based normalization of gene expression data," *Bioinformatics*, vol. 18, pp. 555–565, 2002.

[104] Y. Yang, S. Dudoit, P. Luu, and et al., "Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation," *Nucleic Acids Research*, vol. 30, pp. e15–24, 2002.

[105] J. Quackenbush, "Microarray data normalization and transformation," *Nature Genetics*, vol. Suppl 32, pp. 496–501, 2002.

[106] L. Brás and J. Menezes, "Dealing with gene expression missing data," *IEEE Proc Syst Biol*, vol. 153, pp. 105–119, 2006.

[107] D. Nguyen, N. Wang, and R. Carroll, "Evaluation of missing value estimation for microarray data," *J Data Sci*, vol. 2, pp. 347–370, 2004.

[108] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature-selection," *Pattern Recognition Letter*, vol. 15, pp. 1119–1125, 1994.

[109] C. Sima and E. R. Dougherty, "The peaking phenomenon in the presence of feature selection," *Pattern Recognition Letters*, vol. 29, pp. 1667–1674, 2008.

[110] C. Sima, S. Attoor, U. Braga-Neto, J. Lowey, E. Suh, and E. R. Dougherty, "Impact of error estimation on feature-selection algorithms," *Pattern Recognition*, vol. 38, pp. 2472–2482, 2005.

[111] M. Kudo and J. Sklansky, "Classifier-independent feature selection for two-stage feature selection," in *Lecture Notes in Computer Science, Advances in Pattern Recognition*, A. Amin, H. F. D. Dori, and P. Pudil, Eds., vol. 1451, pp. 548–554. Springer, Berlin, 1998.

[112] L. Veer, H. Dai, M. Vijver, Y. He, A. Hart, M. Mao, H. Peterse, K. Kooy, M. Marton, A. Witteveen, G. Schreiber, R. Kerkhoven, C. Roberts, P. Linsley, R. Bernards, and S. Friend, "Gene expression profling predicts clinical outcome of breast cancer," *Nature*, vol. 415, pp. 530–536, 2002.

[113] M. Vijver, Y. He, L. Veer, H. Dai, A. Hart, D. Voskuil, G. Schreiber, J. Peterse, C. Roberts, M. Marton, M. Parrish, D. Atsma, A. Witteveen, A. Glas, L. Delahaye, T. Velde, H. Bartelink, S. Rodenhuis, E. Rutgers, S. Friend, and R. Bernards, "A gene-expression signa-ture as a predictor of survival in breast cancer," *New Eng J Med*, vol. 347, pp. 1999–2009, 2002.

[114] J. Lapointe, C. Li, J. Higgins, M. Rijn, E. Bair, K. Montgomery, M. Ferrari, L. Egevad, W. Ayford, U. Bergerheim, P. Ekman, A. DeMarzo, R. Tibshirani, D. Botstein, P. Brown, J. Brooks, and J. Pollack, "Gene expression profiling

identifies clinically relevant subtypes of prostate cancer," *Proc Natl Acad Sci*, vol. 101, pp. 811–816, 2004.

[115] L. Weng, H. Dai, Y. Zhan, Y. He, S. Stepaniants, and D. Bassett, "Rosetta error model for gene expression analysis," *Bioinformatics*, vol. 22, pp. 1111–1121, 2006.

VITA

Youting Sun received the B.S. degree in Control Science and Engineering from Tsinghua University, Beijing, China, in 2007. She was then enrolled in the department of Electrical and Computer Engineering at Texas A&M University in College Station as a Ph.D. student, where she joined the Genomic Signal Processing lab under the supervision of Dr. Ulisses Braga-Neto and Dr. Edward. R. Dougherty. She was conferred the Ph.D. degree in May 2012. Her research interests include computational biology, mathematical modeling, and signal processing. Her address is:

GSP Lab

9 Zachry

Texas A&M University

College Station, Texas 77843

The typist for this dissertation was Youting Sun.