Bayesian ABC-MCMC Classification of Liquid Chromatography–Mass Spectrometry Data

Upamanyu Banerjee and Ulisses M. Braga-Neto

Department of Electrical and Computer Engineering, Center for Bioinformatics and Genomics Systems Engineering, Texas A&M University, College Station, TX, USA.

Supplementary Issue: Statistical Systems Theory in Cancer Modeling, Diagnosis, and Therapy

ABSTRACT: Proteomics promises to revolutionize cancer treatment and prevention by facilitating the discovery of molecular biomarkers. Progress has been impeded, however, by the small-sample, high-dimensional nature of proteomic data. We propose the application of a Bayesian approach to address this issue in classification of proteomic profiles generated by liquid chromatography–mass spectrometry (LC-MS). Our approach relies on a previously proposed model of the LC-MS experiment, as well as on the theory of the optimal Bayesian classifier (OBC). Computation of the OBC requires the combination of a likelihood-free methodology called approximate Bayesian computation (ABC) as well as Markov chain Monte Carlo (MCMC) sampling. Numerical experiments using synthetic LC-MS data based on an actual human proteome indicate that the proposed ABC-MCMC classification rule outperforms classical methods such as support vector machines, linear discriminant analysis, and 3-nearest neighbor classification rules in the case when sample size is small or the number of selected proteins used to classify is large.

KEYWORDS: optimal Bayesian classifier, approximate Bayesian computation, Markov chain Monte Carlo, liquid chromatography-mass spectrometry, proteomics

SUPPLEMENT: Statistical Systems Theory in Cancer Modeling, Diagnosis, and Therapy

CITATION: Banerjee and Braga-Neto. Bayesian ABC-MCMC Classification of Liquid Chromatography–Mass Spectrometry Data. *Cancer Informatics* 2015:14(S5) 175–182 doi: 10.4137/CIN.S30798.

TYPE: Methodology

RECEIVED: June 09, 2016. **RESUBMITTED:** August 23, 2016. **ACCEPTED FOR PUBLICATION:** August 28, 2016.

ACADEMIC EDITOR: J. T. Efird, Editor in Chief

PEER REVIEW: Three peer reviewers contributed to the peer review report. Reviewers' reports totaled 495 words, excluding any confidential comments to the academic editor.

FUNDING: Authors disclose no external funding sources.

COMPETING INTERESTS: Authors disclose no potential conflicts of interest.

CORRESPONDENCE: ulisses@ece.tamu.edu

Published by Libertas Academica. Learn more about this journal.

Introduction

Recent advances in high-throughput technologies in proteomics promise to revolutionize cancer treatment and prevention by facilitating the discovery of molecular biomarkers, which can be used to improve diagnosis, guide targeted therapy, and monitor therapeutic response.[1] Among all high-throughput proteomic technologies, mass spectrometry has increasingly become the method of choice for the analysis of complex protein samples.[2] High molecular specificity and excellent detection sensitivity explain the widespread adoption of mass spectrometry (MS)-based proteomics as a popular tool for the identification and quantification of the composition of complex proteome mixtures.

However, to date, the rate of discovery of successful biomarkers is still unsatisfactory. In addition to challenges such as the high dynamic range of proteins[3] and inaccurate protein quantification,[4] an important impediment to progress is that, in clinical applications of mass spectrometry, the number of samples available is extremely small, whereas mass spectra contain hundreds of thousands of intensity measurements with signals generated by thousands of proteins/peptides. 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. Through model-based approaches and simulation using ground-truthed synthetic data, the problem of biomarker discovery can be studied and evaluated.

In this paper, we propose the application of a Bayesian approach to address the small-sample, high-dimensionality problem in the classification of proteomic profiles generated by liquid chromatography–mass spectrometry (LC-MS). Our approach relies on the detailed LC-MS experiment pipeline model developed in Ref. 5, as well as on the theory of the optimal Bayesian classifier (OBC), proposed in Ref. 6. However, the complexity of the LC-MS experiment, involving steps of sample preparation, protein digestion, peptide ionization, peptide detection, and protein quantification, implies that the likelihood function for the LC-MS model is exceedingly complex, requiring the application of a *likelihood-free* Bayesian approach. In this paper, we apply a new likelihood-free methodology called approximate Bayesian computation (ABC).[7] The basic ABC rejection sampling method generates candidate parameters by sampling from the prior distribution and creates a model-based simulated dataset. If the dataset conforms to the observed dataset, the candidate can be retained as a sample from the posterior distribution. Thus, one can avoid evaluating the likelihood function, which is essential for classical Bayesian posterior simulation methods. The ABC approach can also be implemented via a combination of rejection sampling and Markov chain Monte Carlo (MCMC) sampling.[8]

The detailed implementation of our approach involves first the prior calibration of the hyperparameters of the LC-MS model using an ABC approach via rejection sampling and then using the ABC method implemented via an MCMC procedure to obtain samples from the posterior distribution of the protein concentrations, which are used to approximate the OBC using Monte Carlo integration and kernel smoothing. Numerical experiments using synthetic LC-MS data based on an actual human proteome indicate that the ABC-MCMC classification rule outperforms classical methods such as support vector machines (SVMs), linear discriminant analysis (LDA), and 3-nearest neighbor (3NN) classifiers in the case when sample size is small or the number of selected proteins used to classify is large. We also quantify the effect of experimental parameters such as the coefficient of variation (noise) and instrument peptide efficiency factor on classification accuracy.

The paper is organized as follows. The "LC-MS Model" section surveys the LC-MS model proposed in Ref. 5, which is the basis for our inference approach. The "ABC-MCMC Classification Framework" section describes in detail the algorithms for prior calibration, sampling from the posterior, and computation of the ABC-MCMC classifier. The "Numerical Experiments" section presents the results of a numerical experiment using synthetic LC-MS data corresponding to a subset of the human proteome. Finally, the "Conclusion" section brings concluding remarks.

LC-MS Model

Here, we describe briefly the label-free LC-MS model proposed in Ref. 5. Two sample classes are considered, control (class 0) and treatment (class 1). There are n sample profiles from each class, sharing N_{pro} protein species from a specified proteome, which is typically input into the model as a FASTA file. As argued in Ref. 9, protein concentration in the control sample is best described as a Gamma distribution,

$$\gamma_l = \Gamma(k,\theta),\ \ l = 1,2,\ldots,N_{\text{pro}}, \tag{1}$$

where the shape k and scale θ parameters are assumed to be uniform random variables, such that $k \sim \text{Unif}(k_{\text{low}}, k_{\text{high}})$ and $\theta \sim \text{Unif}(\theta_{\text{low}}, \theta_{\text{high}})$. The values for k_{low}, k_{high}, θ_{low}, and θ_{high} were chosen to adequately reflect the dynamic range of protein abundance levels (see the "Numerical Experiments" section).

According to whether there is a significant difference in abundance between control and treatment populations, proteins are divided into biomarker (differentially expressed) proteins and background (not differentially expressed) proteins. The difference in abundance for biomarker proteins is quantified by the fold change,

$$f_l = \begin{cases} a_l, & \text{if the } l\text{th protein is overexpressed,} \\ 1/a_l, & \text{if the } l\text{th protein is underexpressed,} \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

The multivariate Gaussian distribution is recommended as the model for protein concentration variations in each class.[10] Accordingly, the protein expression level for the lth protein in the jth sample profile is modeled as

$$c_{lj}^{\text{pro}} \sim \begin{cases} \mathcal{N}([\gamma_1,\gamma_2,\ldots,\gamma_{N_{\text{pro}}}],\Sigma), & \text{if } j \in \text{ class } 0, \\ \mathcal{N}([\gamma_1 f_1,\gamma_2 f_2,\ldots,\gamma_{N_{\text{pro}}} f_{N_{\text{pro}}}],\Sigma), & \text{if } j \in \text{ class } 1. \end{cases} \tag{3}$$

In this paper, we assume a diagonal covariance matrix $\Sigma = [\sigma_{lk}^2]_{N_{\text{pro}} \times N_{\text{pro}}}$ such that protein concentrations are mutually independent (the results will still be approximately valid as long as the proteins are only weakly correlated):

$$\Sigma = \begin{bmatrix} \sigma_{11}^2 & 0 & \ldots & 0 \\ 0 & \sigma_{22}^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sigma_{N_{\text{pro}}N_{\text{pro}}}^2 \end{bmatrix} \tag{4}$$

where

$$\sigma_{lk}^2 = \begin{cases} \sigma_{ll}^2, & \text{if } l = k \ \text{ and } l,k = 1,2,\ldots,N_{\text{pro}}, \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

and

$$\sigma_{ll}^2 = \varphi \times \gamma_l^2, \qquad l = 1,2,\ldots,N_{\text{pro}}. \tag{6}$$

The coefficient of variation φ is calibrated based on the observed data.

In order to perform *in silico* tryptic digestion of the protein samples, we use the peptide mixture model from openMS.[11] Let Ω_i be the set of all proteins that contain the ith peptide. If there are N_{pep} peptide species, in total, across all proteins in a given sample, then their molar concentrations are given as

$$c_{ij}^{\text{pep}} = \sum_{k \in \Omega_i} c_{kj}^{\text{pro}}, \quad i = 1,2,\ldots,N_{\text{pep}},\ j = 1,\ 2,\ \ldots,\ 2n. \tag{7}$$

In general, ion abundance in MS data bears the signature of the concentration of a peptide type, say i in sample j. Taking measurement uncertainty factors in consideration, one may envisage that the expected readout μ_{ij} of the abundance of said peptide can be modeled as,

$$\mu_{ij} = c_{ij}^{\text{pep}}\, e_i \kappa, \quad i = 1,2,\ldots,N_{\text{pep}}, \quad j = 1,2,\ldots,\ldots 2n, \tag{8}$$

where e_i denotes the peptide efficiency factor and κ represents the LC-MS instrument response factor.[5]

The true peptide abundance differs from its readout due to noise. Accordingly, the actual abundance of a peptide v_{ij} is modeled as $v_{ij} = \mu_{ij} + \epsilon_{ij}$, where ϵ_{ij} is additive Gaussian noise and follows the distribution

$$\epsilon_{ij} \sim \mathcal{N}(0, \alpha\mu_{ij}^2 + \beta\mu_{ij}), \qquad i = 1,2,\ldots,N_{\text{pep}}, \quad j = 1,2,\ldots,2n, \qquad (9)$$

where α and β specify the quadratic dependence of the noise variance on the expected abundance.[5,12]

Peptide signals observed in mass spectra are in fact the result of true signals with interfering noise signals and also signals from other peptides. Therefore, the signal-to-noise ratio (SNR) affects the true positive rate (TPR) greatly. To take account of this, we describe the SNR as

$$\text{SNR} = \frac{E[v]^2}{\text{Var}(v)} = \frac{1}{\alpha + \dfrac{\beta}{\mu}}. \qquad (10)$$

Taking interfering signals in consideration, the TPR of peptides is defined as

$$\text{TPR} = (t \times \text{SNR}^p + b) \times o_{ij}, \qquad (11)$$

where o_{ij} is an overlapping factor. If algorithms like NITPICK, BPDA, and BPDA2d are used, then $o_{ij} \approx 1$.[5]

Finally, we consider in our model three peptide filters, in order: (1) nonunique peptides present in more than one protein of the proteome in study are discarded; (2) peptides with missing value rates greater than 0.7 are discarded; and (3) among the remaining peptides, those having correlation larger than 0.6 with all other peptides are kept.

The MS1 output provides information about detected peptides, their abundances, and related characteristics. The process of filtering these data and compiling the parent protein abundances from the raw peptide data is called *protein abundance roll-up*. To obtain the identities of the parent proteins from captured peptide sequence information, one will often use a second round of MS and search available MS/MS (MS2) databases. Alternatively, the accurate mass and time approach matches peptides to databases using the monoisotopic mass and elution time predictors, obviating the need of a second step of MS.[13] We will assume here that data are available in the form of rolled-up abundances, whereby the readout of protein l in sample j can be written as

$$x_{lj} = \frac{1}{\kappa n_l} \sum_{i \in \mathcal{N}_l} v_{ij}, \qquad l = 1,2,\ldots,N_{\text{pro}}, \quad j = 1,2,\ldots,2n, \qquad (12)$$

where κ is the instrument response factor, N_l is the set of all peptides present in protein l that are retained after the filtering scheme described in the previous paragraph, and n_l is the number of peptides in set N_l. The protein abundance is set to zero when less than two peptides pass the previous filters.

ABC-MCMC Classification Framework

Bayesian analysis for complex models used in recent applications involve intractable likelihood functions, which has prompted the development of new algorithms generally called approximate Bayesian computation (ABC). In this approach, one generates candidate parameters by sampling from the prior distribution and creating a model-based simulated dataset. If the dataset conforms to the observed dataset, the candidate can be retained as a sample from the posterior distribution. Thus, one can avoid evaluating the likelihood function, which is essential for classical Bayesian posterior simulation methods. The ABC approach can be implemented via rejection sampling, MCMC, and sequential Monte Carlo methods.[8] Utilizing the LC-MS proteomics model described in the last section, we first do prior calibration of the hyperparameters using an ABC approach via rejection sampling, and then use the ABC method implemented via an MCMC procedure to obtain samples from the posterior distribution of the protein concentrations in order to derive the ABC-MCMC classifier for LC-MS data.

Overview of the inference procedure. The sample data $S = S_0 \cup S_1$ consist of two subsamples S_0 and S_1, corresponding to the control group (eg, healthy volunteers) and treatment group (eg, cancer patients), respectively, where each subsample contains n protein abundance profiles. Given the sample data, the total number of proteins N_{pro} is reduced via feature selection (eg, ranking by the two-sample t-test statistic) to a tractable number d of selected proteins. According to the adopted LC-MS model, described in the "LC-MS Model" section, the protein abundance profiles are a function of the baseline protein concentration vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_d)$, (b) the *prior hyperparameters k, θ, φ, f*, consisting of shape and scale parameters of the Gamma distribution in (1), the fold change parameters in (2), and the coefficient of variation in (6); and (c) the LC-MS instrument-related parameters κ, α, β, $\mathbf{e}$, b, t, p, which are assumed to be known for a given instrument (see Table 1 for the value of these parameters in our numerical experiment). Figure 1 displays the relationship among these various parameters.

Our approach consists of treating $\boldsymbol{\gamma}$ as the *hidden parameter vector*, posterior samples of which are obtained using an ABC-MCMC sampling method, after a step of calibration of the hyperparameters using ABC rejection sampling. The samples from the posterior allow us to calculate the OBC for the problem. All these steps are described in detail in the sequel.

Algorithm 1 Prior calibration of k, θ, and φ using ABC rejection sampling.

1. Generate M_{cal} triplets of parameters of $\{k^{(t)}, \theta^{(t)}, \varphi^{(t)}\}$ such that,

$k^{(t)} \sim \text{Unif}(k_{\text{low}}, k_{\text{high}})$, $\theta^{(t)} \sim \text{Unif}(\theta_{\text{low}}, \theta_{\text{high}})$,

$\varphi^{(t)} \sim \text{Unif}(\varphi_{\text{low}}, \varphi_{\text{high}})$

for $t = 1,\ldots, M_{\text{cal.}}$

2. Simulate a control sample set $S_0^{(t)}$ of size n for each triplet $\{k^{(t)}, \theta^{(t)}, \varphi^{(t)}\}$, for $t = 1, 2,\ldots, M_{\text{cal.}}$
3. Accept the triplet $\{k^{(t)}, \theta^{(t)}, \varphi^{(t)}\}$ if $\|\boldsymbol{T}(S_0^{(t)} - \boldsymbol{T}(S_0)\| < \in$, for $t = 1,\ldots,M_{\text{cal,}}$ where $\|\cdot\|$ denotes the Euclidean norm and $\boldsymbol{T}$ denotes the vector sample mean.
4. Let $\mathcal{A} = \{\{k^1, \theta^1, \varphi^1\},\ldots,\{k^{n_a}, \theta^{n_a}, \varphi^{n_a}\}\}$ be the set of all accepted triplets. The calibrated k can be approximated as follows

$$k_{\text{cal}} = \int_{k_{\text{low}}}^{k_{\text{high}}} kp\ (k|S_n)dk \approx \frac{1}{n_a}\sum_{a=1}^{n_a} k^a.$$

Similar Monte Carlo integrations are performed to calculate θ_{cal} and φ_{cal}.

Prior calibration via ABC rejection sampling. Calibration of the hyperparameters k, θ, φ, $\boldsymbol{f}$ is accomplished using the ABC rejection sampling method. Unlike Knight et al.[14], who proposed using discarded features to perform prior calibration for an MCMC implementation of the OBC, here we use the selected features, as we need to calibrate the fold change as well, which is specific to each selected protein.

First, we calibrate k, θ, and φ using the control sample only, since these parameters are common across control and treatment populations and $\boldsymbol{f}$ has not been calibrated yet. The procedure used is displayed in Algorithm 1. In this algorithm, $\in$ is the error tolerance. It has been proved[7] that smaller $\in$ gives better approximation of the posterior $p(k|S_n)$. However, this must be balanced against the possibility that $P(\|\boldsymbol{T}(S_0^{(t)}), \boldsymbol{T}(S_0)\| < \in) \approx 0$, which would prevent convergence to the posterior.

Next we calibrate the fold change parameter $\boldsymbol{f} = (f_1,\ldots,f_d)$ for each selected protein. If sample size is large ($n > 50$) then the simple sample estimate

$$f_{l,\text{cal}} = \frac{T_l(S_1)}{T_l(S_0)}, \text{ for } l = 1,\ldots,d, \tag{13}$$

where T_l denotes the sample mean for the lth selected protein only, is fairly accurate, and may be used as the prior calibration. However, for smaller sample sizes, we follow the steps enumerated below in Algorithm 2.

Posterior sampling via an ABC-MCMC procedure. After prior calibration, we would like now to draw samples from the posterior distribution of the protein baseline expression vector $\boldsymbol{\gamma} = (\gamma_1,\ldots,\gamma_d)$, namely, $p(\gamma|\ S_n) \propto p(S_n|\ \gamma)p(\gamma)$, in order to derive the OBC. In our case, no closed-form expressions for either the likelihood function or posterior distribution exist, so Bayesian analysis is performed using an ABC-MCMC procedure, described in Algorithm 3. After a *burn-in* interval of t_s time steps, the Markov chain is assumed to have become stationary, and $\gamma^{(t_s+1)},\ldots,\gamma^{(t_s+M)}$ may be considered to be samples from the baseline expression posterior distribution $p(\gamma|y=0,S_n)$, while $\gamma^{(t_s+1)} f_{cal},\ldots,\gamma^{(t_s+M)} f_{cal}$ (where vector multiplication is defined as componentwise multiplication) may be taken to be samples from the altered expression posterior distribution $p(\gamma|\ y=1,S_n)$.

Algorithm 2 Prior calibration of f_l, $l = 1,\ldots, d$, using ABC rejection sampling.

1. Generate M_{cal} baseline expression values $\gamma_l^{(t)} \sim \Gamma(k_{cal}, \theta_{cal})$ for $t = 1,\ldots,M_{\text{cal}}$.
2. Simulate a control sample $S_0^{(t)}$ of size n using the baseline expression mean $\gamma_l^{(t)}$, for $t = 1,\ldots,M_{\text{cal}}$ (in fact, only the abundances for the lth protein need to be generated).
3. Accept $\gamma_l^{(t)}$ if $|T_l(S_0^{(t)}) - T_l(S_0)| < \in_1$ and $p_l(S_0^{(t)}, S_0) > 1 - \in_2$, where T_l denotes the sample mean and ρ_l denotes the sample correlation for the abundances of the lth protein only.
4. Generate M_{cal} fold change parameters $f_l^{(t)}$ such that If $T_l(S_1)/T_l(S_0) \geq 1$, then $f_l^{(t)} \sim \text{Unif}(\alpha_{\text{low}}, \alpha_{\text{high}})$, If $T_l(S_1)/T_l(S_0) < 1$, then $f_l^{(t)} \sim \text{Unif}(1/\alpha_{\text{high}}, 1/\alpha_{\text{low}})$, for $t = 1,\ldots, M_{\text{cal.}}$
5. Simulate a treatment sample $S_1^{(t)}$ of size n using the altered expression mean $f_l^{(t)}\gamma_l^{(t)}$, for $t = 1, 2,\ldots, M_{\text{cal}}$ (in fact, only the abundances for the lth protein need be generated).
6. Accept $f_l^{(t)}\gamma_l^{(t)}$ if $|T_l(S_1^{(t)}) - T_l(S_1)| < \in_1$ and $\rho_l(S_1^{(t)}, S_1) > 1 - \in_2$.
7. Let n_a^0 be the number of accepted baseline expression means in step 3 and let n_a^1 be the number of accepted altered expression means in step 6. Define
   $\lambda^0 = n_a^0/M_{\text{cal}}$, the rate of acceptance of control means,
   $\lambda^0 = n_a^1/M_{\text{cal}}$, the rate of acceptance of treatment means.
8. If $\lambda^0 > \lambda^1$ then assign $f_{l,\text{cal}} = 1$ (ie, background protein) and return from the algorithm.
9. Otherwise, $f_{\text{cal},l} \neq 1$ (ie, marker protein). For all the accepted altered expression means, we perturb each of the fold changes $f_l^* = f_l + N_l$, where N_l is zero-mean Gaussian noise with a small variance. With these perturbed fold changes, we again apply the ABC rejection algorithm, this time with error tolerances, $\in_1' < \in_1$ and $\in_2' < \in_2$.
10. The mean of all accepted fold change parameters in step 9 is a reasonably accurate fold changed f_{cal} for the given protein.

Optimal Bayesian classifier. Let ψ: $R^d \rightarrow \{0, 1\}$ be a classifier that takes a protein abundance profile $\boldsymbol{X} \in R^d$ into one of the two labels 0 or 1, which code for the control (baseline expression) and treatment (altered expression) populations, respectively. The *error* of the classifier is the probability of a mistake given the sample data:

$$\varepsilon[\psi] = P(\psi(\boldsymbol{X}) \neq Y|S), \tag{14}$$

where $Y \in \{0, 1\}$ denotes the true label corresponding to $\boldsymbol{X}$.

Algorithm 3 Posterior sampling of γ using an ABC-MCMC procedure.

1. Generate $\boldsymbol{\gamma}^{(0)} = (\gamma_0, \gamma_1, \ldots, \gamma_d)$ such that $\gamma_l \sim \Gamma(k, \theta)$, for $l = 1, 2, \ldots, d$.
2. Simulate control and treatment samples $S_0^{(0)}$ and $S_1^{(0)}$ of size n using $\boldsymbol{\gamma}^{(0)}$ and $\boldsymbol{\gamma}^{(0)} \boldsymbol{f}_{\text{cal}}$, respectively (where vector multiplication is defined as componentwise multiplication).
3. Accept $\boldsymbol{\gamma}^{(0)}$ if $\|\boldsymbol{T}(S_0^{(0)} - \boldsymbol{T}(S_0)\| < \epsilon_0$ and $\|\boldsymbol{T}(S_1^{(0)} - \boldsymbol{T}(S_1)\| < \epsilon_1$, otherwise repeat steps 1 and 2 until the condition is met.

For $t = 0,1,\ldots, t_s, t_{s+1},\ldots,t_s + M$ where t_s is the burn-in period, repeat:

4. Generate $\boldsymbol{\gamma}^{(t+1)} \sim g(\boldsymbol{\gamma}; \boldsymbol{\gamma}^{(t)})$, where the proposal density $g(\boldsymbol{\gamma}; \boldsymbol{\gamma}^{(t)})$ is multivariate Gaussian $\mathcal{N}_d(\gamma^{(t)}, \sigma^2 I_d)$, with a small variance σ^2.
5. Simulate control and treatment samples $S_0^{(t+1)}$ and $S_1^{(t+1)}$ of size n using $\boldsymbol{\gamma}^{(\text{t}+1)}$ and $\boldsymbol{\gamma}^{(\text{t}+1)} \boldsymbol{f}_{\text{cal}}$, respectively.
6. Let $q = \begin{cases} \min\left(1, \dfrac{p(\gamma^{(t+1)})g(\gamma^{(t)};\gamma^{(t+1)})}{p(\gamma^{(t)})g(\gamma^{(t+1)};\gamma^{(t)})}\right), & \text{if} \|T(S_0^{(t+1)}) - T(S_0)\| < \epsilon_0 \\ & \text{and} \|T(S_1^{(t+1)}) - T(S_1)\| < \epsilon_1 \\ 0, & \text{otherwise,} \end{cases}$

where $p(\cdot)$ is the Gamma prior for protein baseline expression.

5. Accept $\boldsymbol{\gamma}^{(t+1)}$ with probability q, or let $\boldsymbol{\gamma}^{(t+1)} = \boldsymbol{\gamma}^{(t)}$ with probability $1 - q$.

Now, consider a Bayesian setting, where the joint distribution of $(\boldsymbol{X},\boldsymbol{Y})$ depends on a random parameter vector $\boldsymbol{\theta}$. In this case, the classification error $\varepsilon_{\boldsymbol{\theta}}[\psi]$ also becomes a random variable, as a function of $\boldsymbol{\theta}$. The expected value of the classification error over the posterior distribution of $\boldsymbol{\theta}$ becomes the quantity of interest:

$$E_{\theta|S}[\varepsilon_\theta[\psi]] = E_{\theta|S}[P(\psi(X) \neq Y \mid \theta, S)]. \tag{15}$$

The OBC[6] is the classifier that minimizes the quantity in (15):

$$\psi_{\text{OBC}} = \arg\min_{\psi \in \mathcal{C}} E_{\boldsymbol{\theta}|S}[\varepsilon_{\boldsymbol{\theta}}[\psi]], \tag{16}$$

where $\mathcal{C}$ is the space of classifiers. It was shown in Ref. 6 that the OBC is given by

$$\psi_{\text{OBC}}(\boldsymbol{x}) = \begin{cases} 1, & \text{if } E[c|S]p(\boldsymbol{x}|Y=1,S) > (1 - E[c|S])p(\boldsymbol{x}|Y=0,S), \\ 0, & \text{otherwise,} \end{cases} \tag{17}$$

where $c = P(Y = 1 \mid \boldsymbol{\theta})$ is the prior probability of class 1, and

$$p(\boldsymbol{x}|Y=y,S) = \int_\Theta p(\boldsymbol{x}|\boldsymbol{\theta}, Y=y, S)\, p(\boldsymbol{\theta}|Y=y,S)\, d\boldsymbol{\theta}, \quad y = 0,1, \tag{18}$$

are the *effective class-conditional densities*.

In the present case of the LC-MS model discussed in the "LC-MS Model" section, the random parameter vector $\boldsymbol{\theta}$ corresponds to the baseline expression vector $\boldsymbol{\gamma}$. We approximate the integral in (18) using the MCMC samples $\gamma^{(t_s+1)},\ldots,\gamma^{(t_s+M)}$ from the posterior distribution of γ, obtained with Algorithm 3:

$$p(x|Y=y,S) \approx \frac{1}{M} \sum_{t=t_s+1}^{t_s+M} p(x|\gamma^{(t)}, Y=y, S), \quad y = 0,1. \tag{19}$$

Now, the densities $p(\boldsymbol{x}|\boldsymbol{\gamma}^{(t)}, Y = y, S)$, $y = 0, 1$, cannot be directly determined for the LC-MS model, and hence we approximate them using a kernel-based approach. For each MCMC sample $\boldsymbol{\gamma}^{(t)}$, we simulate control and treatment samples $S_0^{(t)}$ and $S_1^{(t)}$ of size n based on $\boldsymbol{\gamma}^{(t+1)}$ and $\boldsymbol{\gamma}^{(t+1)} \boldsymbol{f}_{\text{cal}}$, respectively. Let $S_0^{(t)} = \{\boldsymbol{x}_1^{(t)},\ldots,\boldsymbol{x}_n^{(t)}\}$ and $S_1^{(t)} = \{\boldsymbol{x}_{n+1}^{(t)},\ldots,\boldsymbol{x}_{2n}^{(t)}\}$. Then

$$p(x|\gamma^{(t)}, Y=y, S) \approx \frac{1}{n} \sum_{j=ny+1}^{ny+n} \frac{1}{h^d} K\left(\frac{x - x_j^{(t)}}{h}\right), \quad y = 0,1, \tag{20}$$

where $\boldsymbol{K}$ is a zero-mean, unit-covariance, multivariate Gaussian density, and $h > 0$ is a suitable kernel bandwidth parameter.

In addition, we will assume c to be known (eg, from epidemiological data) and fixed, so $E[c \mid S] = c$. After some simplification, the resulting OBC, which we call an ABC-MCMC Bayesian classifier, is a kernel-based classifier given by

$$\psi_{\text{ABC-MCMC}}(\boldsymbol{x}) = \begin{cases} 1, & \text{if } c \displaystyle\sum_{t=t_s+1}^{t_s+M} \sum_{j=1}^{n} \boldsymbol{K}\left(\frac{\boldsymbol{x} - \boldsymbol{x}_j^{(t)}}{h}\right) \\ & \quad > (1-c) \displaystyle\sum_{t=t_s+1}^{t_s+M} \sum_{j=n+1}^{2n} \boldsymbol{K}\left(\frac{\boldsymbol{x} - \boldsymbol{x}_j^{(t)}}{h}\right), \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

Numerical Experiments

We demonstrate the application of the proposed ABC-MCMC classification rule to synthetic LC-MS data generated from a subset of the human proteome, containing around 4000 drug targets, which was compiled as a FASTA file from DrugBank[15] – this is the same proteome that was used in the numerical experiments of Ref. 5 – and compare its performance against that of popular classification rules: linear support vector machines, LDA, and 3NN.[16] As our interest is on small-sample performance, we selected methods that are simple and known to perform well with small samples and avoid overfitting: linear SVMs are sophisticated methods widely used in the pattern recognition and machine learning communities, which displays minimal overfitting, while LDA and 3NN are classical methods that are well known to have superior small-sample performance.[17]

We select randomly among these data 500 proteins to play the role of background proteins, along with 20 proteins to serve as biomarkers. Synthetic LC-MS protein abundance data were generated using realistic sample preparation, LC-MS instrument characteristics, and protein quantification parameters – see Table 1. These are the "LC-MS experiment parameters" of Figure 1, which are assumed to be known and are held constant throughout the simulation. (For the peptide efficiency factor, values uniformly distributed in the indicated range are randomly generated for each peptide, and then held constant.) As argued in Ref. 5, the values and ranges adopted in Table 1 adequately represent the peptide mixture, peptide abundance mapping, peptide detection and identification, and protein abundance roll-up that is typical in an LC-MS workflow.

The hyperparameter priors for k, θ, φ, f are the uniform distributions shown in Table 2 (except where noted below). The lower and upper bounds of each interval are chosen while keeping in consideration that, in practice, the dynamic range of protein expression level has approximately 4 orders of magnitude.[5] The synthetic sample data were generated using the middle point of each interval as parameters: $k = 2$, $\theta = 1000$, $\varphi = 0.4$, and $\alpha_l = 1.55$ (again, except where noted below).

We consider sample sizes from n = 10 through n = 50 per class, and select d = 3, 5, 8, or 10 proteins from the original 520 proteins using the two-sample t-test (notice that background proteins could be erroneously selected by the t-test, especially for small sample sizes, which makes the experiment realistic).

Table 1. LC-MS parameters used in the experiment.

PARAMETER	SYMBOL	VALUE/RANGE
Instrument response	κ	5
Noise severity	α, β	0.03, 3.6
Peptide efficiency factor	e_i	[0.1–1]
Peptide detection algorithm	b, t, p	0,0.0016,2

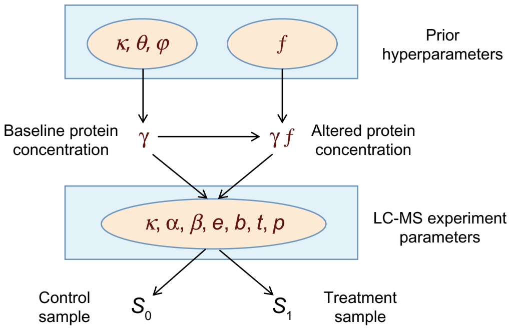
κ, θ, φ
f
Prior hyperparameters
Baseline protein concentration
γ
γ f
Altered protein concentration
κ, α, β, e, b, t, p
LC-MS experiment parameters
Control sample
S0
S1
Treatment sample

Figure 1. Relationship among all parameters of the LC-MS model (see text).

Table 2. Hyperparameter priors used in the experiment.

PARAMETER	SYMBOL	RANGE/VALUE
Shape (gamma distribution)	k	Unif(1.6, 2.4)
Scale (gamma distribution)	θ	Unif(800, 1200)
Coefficient of variation	φ	Unif(0.3, 0.5)
Fold change	α_l	Unif(1.5, 1.6)

For the MCMC step, M = 10,000 samples were drawn from the posterior distribution of $\boldsymbol{\gamma}$, after a burn-in stage of t_s = 3000 iterations, which confers a high degree of accuracy to the approximation. A constant value $c = 0.5$ was assumed in (21).

A total of 12 runs of the experiment were run for each combination of sample size, dimensionality, and parameter settings, and the average true error rate for each classification rule was obtained using a large synthetic test set containing 1000 sample points. This is a comprehensive simulation, given the relatively large computational burden required for accurate prior calibration and ABC-MCMC computation.

The root mean square error (RMS) of the test set error estimator, which reflects the expected distance between the estimate and the true error, is bounded by equation (2.29) in Ref. 17 as follows

$$\text{RMS} \leq \frac{1}{2\sqrt{m}}, \tag{22}$$

where m is the number of test points. With m = 1000, we obtain RMS ≤ 0.016, which is of the order of the differences in average error rates observed in the plots. While not implying statistical significance, this result means that we can assign a large degree of confidence to the comparative results.

Effect of sample size. Figure 2 displays the expected error rates of the various classification rules for varying sample size and fixed number of selected proteins $d = 8$. We can see that, as expected, the expected error rates of all classifiers tend to go down as sample size increases, but the ABC-MCMC classifier has the smallest expected error at small sample sizes. This is in agreement with the predicted superiority of the Bayesian approach in small-sample scenarios. Though the difference in performance among the classification rules may seem to be small, the point to be emphasized is that the

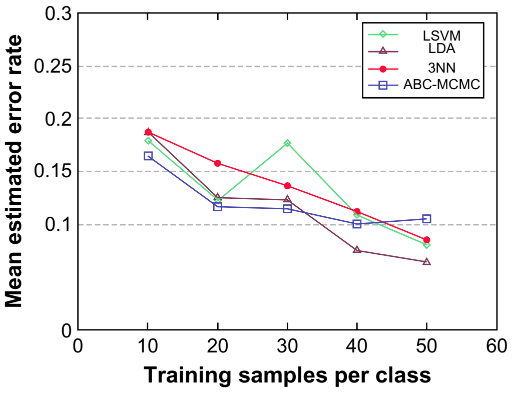

Figure 2. Expected classification error rates for varying sample size and fixed number of selected proteins $d = 8$.

ABC-MCMC displays a consistently smaller error rate for small sample sizes.

Effect of dimensionality. Figure 3 displays the expected error rates of the various classification rules for varying number of selected proteins and fixed sample size $n = 10$ per class. Here we can see that, as the number of selected proteins increases, expected classification error rates tend to go down at first, but then increase slightly, which is in agreement with the well-known *peaking phenomenon* of classification.[18] We can see that the ABC-MCMC classification rule displays the smallest expected error rate when d is large, which once again agrees with the prediction that Bayesian methods perform comparatively well under small-sample scenarios (here, small n/d ratio).

Effect of coefficient of variation. Here we keep both the sample size and the dimensionality fixed at $n = 10$ per class and $d = 8$, respectively, and investigate the impact on classification error rate of an increased variability in the *true* protein concentration values, by changing the value of the coefficient of variation φ used to generate the LC-MS data. To accommodate this change, the hyperparameter prior for φ is changed from the value displayed in Table 2 to Unif($\varphi_0 - 0.1$, $\varphi_0 + 0.1$), where φ_0 is the value used to generate the data. Increasing the coefficient of variation corresponds to the effect of very noisy background proteins in the LC-MS channel. Accordingly, it can be seen in Figure 4 that as φ increases the expected error rates for all classification rules approach the no-information value 0.5, ie, the same error rate of flipping a coin. However, the expected error rate of the ABC-MCMC classification rule approaches 0.5 error rate rather more slowly than the others, indicating superiority in classifying noisy data.

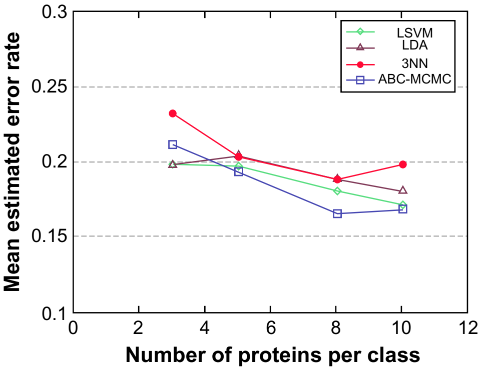

Figure 3. Expected classification error rates for varying number of selected proteins and fixed sample size $n = 10$ per class.

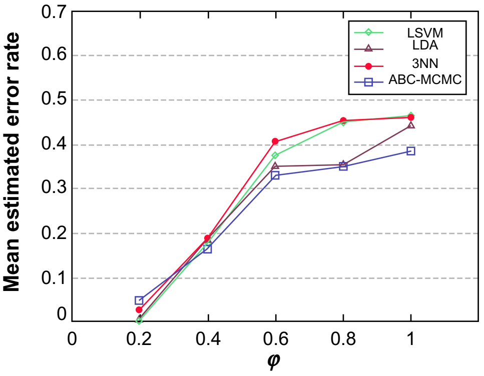

Figure 4. Expected classification error rates for fixed sample size $n = 10$ per class, fixed number of selected proteins $d = 8$, and varying coefficient of variation φ.

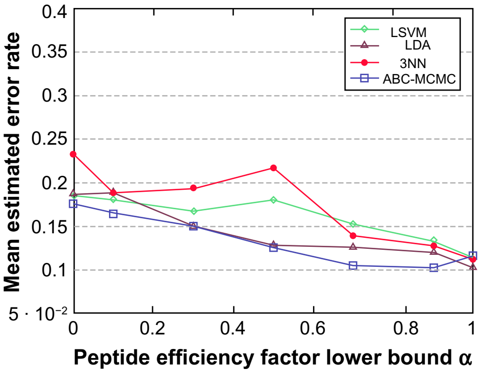

Figure 5. Expected classification error rates for fixed sample size $n = 10$ per class, fixed number of selected proteins $d = 8$, and varying lower bound a for the peptide efficiency factor $e_i \sim$ Unif(α, 1).

Effect of peptide efficiency factor. Finally, we investigate the impact of varying the peptide efficiency factor on the classification error rates. We do this by changing the lower bound in the range for e_i displayed in Table 1 from $\alpha = 0.1$ to a value varying between 0 and 1. The peptide efficiency factor affects how many ions an instrument can detect for a given peptide. Larger values for e_i imply a smaller transmission loss for the corresponding peptide. Increasing the lower bound a will uniformly increase efficiency for all peptides, which corresponds to a better LC-MS instrument. We can see in Figure 5 that, indeed, the expected classification error rates tend to decrease with an increasing lower bound on the peptide efficiency factor, though somewhat modestly (all other things being equal). We can also observe that among all algorithms, the ABC-MCMC classification rule displays the smallest error rate over nearly the entire range in the plot.

Conclusion

We proposed in this paper a model-based Bayesian approach for classification of LC-MS proteomics data with the ultimate goal of facilitating biomarker discovery for cancer research. Our approach combines state-of-the-art Bayesian computation techniques, namely, ABC and MCMC, for the calculation of the OBC. As expected, the proposed Bayesian classifier outperforms other approaches when sample size is small or the number of selected proteins to classify is large. We believe that our simulation using a subset of 4000 human protein drug targets and realistic parameter settings is indicative of the performance of the proposed methodology on real data. The challenges associated with designing experiments and obtaining appropriate real data to calibrate and validate the methodology go beyond the scope of the present paper and are intended to be part of future work.

Author Contributions

Conceived and designed the experiments: UB, UBN. Analyzed the data: UB. Wrote the first draft of the manuscript: UB. Contributed to the writing of the manuscript: UBN. Agree with manuscript results and conclusions: UB, UBN. Made critical revisions and approved final version: UB, UBN. Both authors reviewed and approved of the final manuscript.

REFERENCES

1. Rifai N, Gillette M, Carr SA. Protein biomarker discovery and validation: the long and uncertain path to clinical utility. *Nat Biotechnol*. 2006;24:971–83.
2. Aebersold R, Mann M. Mass spectrometry-based proteomics. *Nature*. 2003;422(6928):198–207.
3. Httenhain R, Malmstrm J, Picotti P, Aebersold R. Perspectives of targeted mass spectrometry for protein biomarker verification. *Curr Opin Chem Biol*. 2009;13:518–25.
4. Griffin N, Yu J, Long F, et al. Label-free, normalized quantification of complex mass spectrometry data for proteomics analysis. *Nat Biotechnol*. 2010;28:83–9.
5. Sun Y, Braga-Neto U, Dougherty E. A systematic model of the LC-MS proteomics pipeline. *BMC Genomics*. 2011;13:S2.
6. Dalton L, Dougherty E. Optimal classifiers with minimum expected error within a Bayesian framework – part I: discrete and Gaussian models. *Pattern Recognit*. 2013;46(5):1301–14.
7. Sisson S, Fan Y. Likelihood-free Markov chain Monte Carlo. In: Brooks S, Gelman A, Jones G, Meng X-L, eds. *Handbook of Markov Chain Monte Carlo*. Boca Raton, FL: Chapman and Hall/CRC Press; 2010:319–41.
8. Peters G, Fan Y, Sisson S. *On Sequential Monte Carlo, Partial Rejection Control and Approximate Bayesian Computation*. Kensington: University of New South Wales; 2009:arxiv:0808.3466v2.
9. Taniguchi Y, Choi PJ, Li GW, et al. Quantifying *E. coli* proteome and transcriptome with single-molecule sensitivity in single cells. *Science*. 2010;329:533.
10. Lu P, Vogel C, Wang R, Yao X, Marcotte EM. Absolute protein expression profiling estimates the relative contributions of transcriptional and translational regulation. *Nat Biotechnol*. 2007;25:117–24.
11. Sturm M, Bertsch A, Gröpl C, et al. OpenMS—an open-source software framework for mass spectrometry. *BMC Bioinformatics*. 2008;9:163.
12. Anderle M, Roy S, Lin H, Becker C, Joho K. Quantifying reproducibility for differential proteomics: noise analysis for protein liquid chromatography-mass spectrometry of human serum. *Bioinformatics*. 2004;20(18):3575–82.
13. Pasa-Tolic L, Masselon C, Barry R, Shen Y, Smith R. Proteomic analyses using an accurate mass and time tag strategy. *Biotechniques*. 2004;37(4): 621–624,626–33,636assim.
14. Knight J, Ivanov I, Dougherty E. MCMC implementation of the optimal Bayesian classifier for non-Gaussian models: model-based RNA-Seq classification. *BMC Bioinformatics*. 2014;15:401.
15. Knox C, Law V, Jewison T. DrugBank 3.0: a comprehensive resource for 'omics' research on drugs. *Nucleic Acids Res*. 2011;39:D1035–41.
16. Webb A. *Statistical Pattern Recognition*. 2nd ed. New York: John Wiley & Sons; 2002.
17. Braga-Neto U, Dougherty E. *Error Estimation for Pattern Recognition*. New York: Wiley; 2015.
18. Hughes G. On the mean accuracy of statistical pattern recognizers. *IEEE Trans Inf Theory*. 1968;IT-14(1):55–63.