Bayesian Classification of Proteomics Biomarkers from Selected Reaction Monitoring Data using an Approximate Bayesian Computation-Markov Chain Monte Carlo Approach

Cancer Informatics
Volume 17: 1–7

DOI: 10.1177/1176935118786927

Kashyap Nagaraja and Ulisses Braga-Neto

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

ABSTRACT: Selected reaction monitoring (SRM) has become one of the main methods for low-mass-range–targeted proteomics by mass spectrometry (MS). However, in most SRM-MS biomarker validation studies, the sample size is very small, and in particular smaller than the number of proteins measured in the experiment. Moreover, the data can be noisy due to a low number of ions detected per peptide by the instrument. In this article, those issues are addressed by a model-based Bayesian method for classification of SRM-MS data. The methodology is likelihood-free, using approximate Bayesian computation implemented via a Markov chain Monte Carlo procedure and a kernel-based Optimal Bayesian Classifier. Extensive experimental results demonstrate that the proposed method outperforms classical methods such as linear discriminant analysis and 3NN, when sample size is small, dimensionality is large, the data are noisy, or a combination of these.

KEYWORDS: Proteomics, biomarker, approximate Bayesian computation (ABC), Markov chain Monte Carlo (MCMC), Optimal Bayesian Classifier (OBC), selected reaction monitoring (SRM)

RECEIVED: February 3, 2018. **ACCEPTED:** May 24, 2018.

TYPE: Review

FUNDING: The author(s) received no financial support for the research, authorship, and/or publication of this article.

DECLARATION OF CONFLICTING INTERESTS: The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

CORRESPONDING AUTHOR: Ulisses Braga-Neto, Department of Electrical & Computer Engineering and Center for Bioinformatics and Genomic Systems Engineering, Texas A&M University, College Station, TX 77843-3128, USA. Email: ulisses@ece.tamu.edu

Introduction

Proteomics is the field which deals with the study of cellular behavior and human disease at the protein level. Recently, cancer treatment and prevention have made great strides, thanks to the development of high-throughput technologies in proteomics. Among these, mass spectrometry (MS) analysis has become the preferred choice because of advantages such as high molecular specificity and better detection sensitivity.[1] Hence, MS is widely used in identification and quantification of complex proteome mixtures with the goal of discovering biomarkers, ie, molecular markers for disease.[2–4]

However, a major challenge in biomarker discovery is the identification of low-abundance proteins in peripheral blood. Selected reaction monitoring (SRM), conducted using a triple-quadrupole (QQQ) instrument, has an extended mass range and has become one of the main methods for low-mass-range–targeted proteomics by MS.[5]

Nevertheless, in most SRM-MS biomarker validation studies, the sample size is very small due to the economic cost of the experiments and difficulty in recruiting cases. Typically, the number of features (measured proteins) is vastly larger than the sample size. Moreover, depending on the instrument sensitivity, the data can be noisy due to low peptide efficiency, ie, low number of ions detected per peptide.

All the aforementioned issues create a difficult challenge to classical data-driven classification methods. In this article, this is addressed by a model-based Bayesian method for classification of SRM-MS data. We perform Bayesian inference of the parameters of the SRM model proposed in the work by Atashpaz-Gargari et al[5] and build a kernel classifier, similar to the classifier for liquid chromatography-mass spectrometry (LC-MS) data proposed in the work by Banerjee and Braga-Neto.[6] As in the latter reference, our method uses a likelihood-free approach, called approximate Bayesian computation (ABC),[7–9] which is necessary because the SRM model of Atashpaz-Gargari et al[5] is complex and does not have an analytical formulation of the likelihood. After calibration of the parameters, the ABC method is implemented via a Markov chain Monte Carlo (MCMC) procedure[10,11] to obtain a sample from the posterior distribution of the protein concentrations. Small MCMC sample sizes are sufficient to obtain a kernel-based implementation of the Optimal Bayesian Classifier (OBC).[12] Extensive experimental results examining the effect of various parameters demonstrate that the proposed method outperforms classical methods such as linear discriminant analysis (LDA) and 3NN,[13] when sample size is very small, dimensionality is large, the data are noisy, or a combination of these.

The organization of the article is as follows. Section "SRM-based MS model" surveys the SRM-MS model. Section "ABC-MCMC classification algorithm" explains in detail the ABC rejection algorithm and the approximate Bayesian computation-Markov chain Monte Carlo (ABC-MCMC) classifier. Section "Numerical experiments and results" presents the numerical results. Section "Conclusions" presents concluding remarks.

Table 1. Parameters used in the experiment.

PARAMETER	SYMBOL	VALUE/RANGE
Instrument response factor	κ	5
Noise severity	α, β	0.03, 3.6
Peptide efficiency factor	e_i	[0.1, 1]
Shape (gamma distribution)	k_a, k_c	Unif(1.6, 2.4), Unif(4, 6)
Scale (gamma distribution)	θ_a, θ_c	Unif(9e6, 11e6), Unif(90, 110)
Purification	η_i	10^{-6}
Coefficient of variation	ϕ	Unif(0.3, 0.5)
Fold change	f	Unif(1.5, 1.6)

SRM-BASED MS MODEL

In this article, we employ the model for the SRM pipeline proposed in the work by Atashpaz-Gargari et al.[5] Next, we review briefly each of the main components of this model.

Protein mixture model

The protein mixture model concerns the true abundance of proteins in the SRM experiment. There are n samples in each class; for convenience, the 2 classes are labeled as 0 for control and 1 for treatment. There are N^a_{pro} proteins, N^c_{pro} of which are low-abundance candidates for biomarker validation. Protein identities are input as a FASTA file. As argued in previos works,[5,14] protein concentration can be modeled by a gamma distribution. Hence, the protein concentration is given by

$$\gamma_i \sim \begin{cases} \Gamma(k_c, \theta_c) & i = 1, 2, 3, \ldots, N^c_{pro} \\ \Gamma(k_a, \theta_a) & i = N^c_{pro} + 1, N^c_{pro} + 2, \ldots, N^a_{pro} \end{cases} \quad (1)$$

The variables k and θ are, respectively, shape and scale parameters. These are uniform random variables defined as $k_c \sim \text{Unif}(k_c^{low}, k_c^{high})$, $k_a \sim \text{Unif}(k_a^{low}, k_a^{high})$ and $\theta_c \sim \text{Unif}(\theta_c^{low}, \theta_c^{high})$, $\theta_a \sim \text{Unif}(\theta_a^{low}, \theta_a^{high})$, respectively. The initial values of these variables, which are displayed in Table 1, reflect the dynamic range of protein abundance levels while taking into account that the candidate proteins are expressed at a much lower level than the background proteins. The initial values used here are consistent with values obtained experimentally in the work by Taniguchi et al[14] as well as the hyperparameter values used in the work by Atashpaz-Gargari et al.[5] Furthermore, these initial values are modified based on the data, as part of the prior calibration process described in Algorithm 1.

Proteins are divided into biomarker (differentially expressed) and nonbiomarker (not differentially expressed) proteins. We use fold change to quantify the difference:

$$f_l = \begin{cases} a_i & \text{if the protein i is overexpressed} \\ \dfrac{1}{a_i} & \text{if the protein is underexpressed} \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

for $l = 1, \ldots, N^a_{pro}$. The fold change parameter a_i is uniformly distributed in the interval $[1, h]$, for $h > 1$. The value of h used here is displayed in Table 1.

While the gamma distribution is chosen for mean protein concentrations, the variation of protein concentration is modeled by a multivariate gaussian vector. Accordingly, the concentration of protein l in class j is modeled as follows:

$$C^{pro}_{lj} \sim \begin{cases} N([\gamma_1, \gamma_2, \ldots, \gamma_{N^a_{pro}}]), \Sigma) & \text{for } j \in \text{class } 0 \\ N([\gamma_1 f_1, \gamma_2 f_2, \ldots, \gamma_{N^a_{pro}} f_{N^a_{pro}}]), \Sigma) & \text{for } j \in \text{class } 1 \end{cases} \quad (3)$$

for $l = 1, \ldots, N^a_{pro}$. Here, we consider a diagonal covariance matrix $\Sigma = [\sigma^2_{lk}]_{N_{pro} X N_{pro}}$ so that the protein concentrations are mutually independent or very weakly correlated (correlation between proteins can be included at the cost of adding more parameters to the model):

$$\Sigma = \begin{bmatrix} \sigma^2_{11} & 0 & 0 & \ldots & 0 \\ 0 & \sigma^2_{22} & 0 & \ldots & 0 \\ . & . & . & \ldots & . \\ . & . & . & \ldots & . \\ 0 & 0 & 0 & \ldots & \sigma^2_{N^a_{pro}} \end{bmatrix} \quad (4)$$

where

$$\sigma^2_{ij} = \begin{cases} \sigma^2_{ii} & \text{if } i = j \text{ and } i, \quad j = 1, \ldots, N^a_{pro} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

and

$$\sigma^2_{ii} = \phi * \gamma^2_i, \quad i = 1, \ldots, N^a_{pro} \quad (6)$$

The coefficient of variation ϕ has the initial value displayed in Table 1, which is the same as the one used in the work by Banerjee and Braga-Neto.[6] This value is modified based on the data, as part of the prior calibration process described in Algorithm 1.

To model the purification process usually performed as part of the SRM-MS protocol, we select a set G_p of high-abundance proteins to be removed (in fact, attenuated) from the protein mixture:

$$\hat{C}^{pro}_{ij} = \begin{cases} \eta_i C^{pro}_{ij} & \text{for } i \in G_p \\ C^{pro}_{ij} & \text{otherwise} \end{cases} \quad (7)$$

The value for η_i corresponds to the efficiency of the purification process and should be very small. The value assumed here is displayed in Table 1.

Peptide mixture model

In SRM-MS, tryptic digestion of proteins is performed to generate small-mass peptides. Let Ω_i be the set of all the proteins which contain the ith peptide:

$$C_{ij}^{pep} = \sum_{k \in \Omega_i} \hat{C}_{kj}^{pro} \quad i = [1,2,\ldots N_c^{pp}],\ j \in [0,1] \tag{8}$$

The readout abundance μ_{ij} of the peptide can be modeled as follows:

$$\mu_{ij} = C_{ij}^{pep} e_i \kappa \tag{9}$$

Here, e_i represents the peptide efficiency factor and κ represents the LC-MS response factor.

However, the true peptide abundance is different from its readout value due to the noise:

$$v_{ij} = \varepsilon_{ij} + \lambda_{ij} \quad i = [1,2,\ldots,N_c^{pp}],\ j \in [0,1] \tag{10}$$

where ε_{ij} is additive gaussian noise, which has a quadratic dependence on μ_{ij} s given below:

$$\varepsilon_{ij} \sim N(0, \alpha\mu_{ij}^2 + \beta\mu_{ij}) \quad i = [1,2,\ldots,N_c^{pp}],\ j \in [0,1] \tag{11}$$

where λ_{ij} is the additive exponential noise introduced due to transition effects:

$$\lambda_{ij} \sim \mathrm{Exp}(\mu_{tran} \times \mu_{ij}) \tag{12}$$

where μ_{tran} is a fixed constant.

The next step is called *protein abundance roll-up*. This is the process of obtaining the abundances of the parent proteins from the abundances and related characteristisc of their child peptides, detected during the MS1 process. To obtain the identities of the parent proteins, a second round of MS, called MS/MS, is often used and available databases of identities are searched. Here, we assume that the data from the rolled up abundances can be obtained and the readout of protein l in sample j is given by

$$x_{lj} = \frac{1}{\kappa\eta_l} \sum_{i \in N_l} v_{lj} \quad l = [1,2,\ldots,N_{pro}],\ j \in [0,1] \tag{13}$$

where κ is the instrument response factor, N_l is the set of proteins present in peptide l and η_l is the number of peptides in set N_l. The data x_{lj} in equation (13) are then used for classification.

ABC-MCMC Classification Algorithm

As described in the introduction section, the algorithm mainly has 3 steps: prior calibration via ABC rejection sampling, posterior sampling using an ABC-MCMC algorithm, and classification using a kernel-based method. We describe each of these steps below.

Prior calibration via ABC rejection sampling

Once the protein abundances as described in equation (7) are obtained, the total number of proteins N_{pro}^a is reduced via a feature selection algorithm. As per the equations in the previous section, the protein abundance profiles are a function of the following:

- Baseline parameters $\gamma = [\gamma_1, \gamma_2, \ldots, \gamma_d]$
- Prior hyperparameters: $k_a, k_c, \theta_a, \theta_c, \phi, f$
- Instrument parameters: $\kappa, \alpha, \beta, e_i$

ALGORITHM 1. PRIOR CALIBRATION OF $k_c, k_a, \theta_c, \theta_a, \phi$ USING ABC REJECTION SAMPLING

1. Generate M_{cal} quintuplets of parameters of $k_c, k_a, \theta_c, \theta_a, \phi$ such that

   $k_a^{(t)} \sim Unif(k_a^{low}, k_a^{high})$
   $k_c^{(t)} \sim Unif(k_c^{low}, k_c^{high})$
   $\theta_a^{(t)} \sim Unif(\theta_a^{low}, \theta_a^{high})$
   $\theta_c^{(t)} \sim Unif(\theta_c^{low}, \theta_c^{high})$
   $\phi^{(t)} \sim Unif(\phi^{low}, \phi^{high})$
   for $t = 1,2,\ldots,M_{cal}$
2. Now simulate a control sample set $S_0^{(t)}$ of size n for each quintuplet of parameters **for** $t = 1, 2, \ldots, M_{cal}$
3. Accept the quintuplet $(k_a^{(t)}, k_c^{(t)}, \theta_a^{(t)}, \theta_c^{(t)}, \phi_a^{(t)})$ if $\| \mathbf{T}(S_0^{(t)}) - \mathbf{T}(S_0) \| < \varepsilon$, for $t = 1,2,\ldots,M_{cal}$. Here, $\|\|$ denotes the Euclidean norm and **T** denotes vector sample mean.
4. Let $B = [(k^1, \theta^1, \phi^1), \ldots, (k^{n_a}, \theta^{n_a}, \phi^{n_a})]$ be the set of accepted triplets.
5. The calibrated k can be approximated as follows:

$$k_a^{cal} = \int_{k_a^{low}}^{k_a^{high}} k_p(k_a \mid S_n) dk = \frac{1}{n_a} \sum_{a=1}^{n_a} k_a^{cal}$$

6. Similarly, other 4 parameters are also calculated.

Prior calibration via ABC rejection sampling is as described in Algorithm 1. Monte Carlo integrations are performed to obtain a set of parameters and only some of them are kept and rest are rejected via comparing with a threshold. All the approximated triplets are averaged to obtain the optimal parameter.

In this algorithm, ε is the error tolerance. This has to be chosen optimally so that it should not be too high for bad samples to be accepted or it should not be very small that all the samples are accepted, ie, $P(||T(S_0^{(t)}), T(S_0)|| < \varepsilon) \approx 0$

Once the optimal parameters are obtained, the fold change vector is calculated by the following sample mean estimate:

$$f_{l,cal} = \frac{T_l(S_1)}{T_l(S_0)}, \quad l = 0,1,2,\ldots,d \tag{14}$$

where T_l denotes the lth sample mean for the selected protein only.

ABC-MCMC posterior sampling

ABC-MCMC sampling is as described in Algorithm 2. Vector $\gamma = \gamma_1, \gamma_2, \ldots, \gamma_d$ is sampled from $p(\gamma \mid S_n) \propto p(S_n \mid \gamma) p(\gamma)$. After a burn-in period for the Markov chain of t_s, the next M samples from t_s to $t_s + M$ are considered as the generated data. Proper selection of the thresholds in step 4 of Algorithm 2 plays a very important role in the performance of the ABC-MCMC algorithm.

ALGORITHM 2. OBTAIN THE POSTERIOR SAMPLES OF γ USING ABC-MCMC ALGORITHM

1. Generate the mean vector $\gamma^{(0)} = (\gamma_0, \gamma_1, \ldots, \gamma_d)$ from the Γ distribution with optimal parameters generated in Algorithm 1.
   For $t = 0,1,\ldots,t_s,t_{s+1},\ldots,t_s + M$ where t_s is the burn-in period do:
2. Generate $\gamma^{(t+1)} = \textbf{\textit{ColMeans}}(S_0^{(t)})$ where *ColMeans* is a function which calculates mean feature (protein) wise.
3. Simulate the control and treatment samples S_0^{t+1} and S_1^{t+1} each of size using $\gamma^{(t+1)}$ and $\gamma^{(t+1)} \cdot f_{cal}$, respectively.
4. Let
   $$q = \begin{cases} 1 & \| \mathbf{T}(S_0^{(t+1)}) - \mathbf{T}(S_0) \| < \epsilon_0 \text{ and } \| \mathbf{T}(S_1^{(t+1)}) - \mathbf{T}(S_1) \| < \epsilon_1 \\ 0 & \textit{otherwise} \end{cases}$$
5. If $q = 1$, accept $\gamma^{(t+1)}$ else $\gamma^{(t+1)} = \gamma^{(t)}$

Kernel-based classification

We employ the kernel-based scheme proposed in the work by Banerjee and Braga-Neto,[6] which is itself based on the OBC in Dalton and Dougherty.[12] One of the issues with kernel-based classification is choosing the right value of the kernel bandwidth parameter. If the value of the bandwidth parameter chosen is high, then it leads to oversmoothing and thus hiding many details in the data distribution. However, a small value for the bandwidth parameter leads to undersmoothing and thus many spurious noisy elements in the data are not eliminated. To address this, we employ an ensemble method, where different classifiers with different bandwidth parameters are obtained and then majority vote is used for classification. The classification algorithm is described in detail in Algorithm 3.

ALGORITHM 3. USING THE ABC-MCMC–BASED POSTERIOR SAMPLES FOR CLASSIFICATION.

1. Choose a set of kernel bandwidth parameters $h = (h_1, h_2, \ldots, h_f)$ where f is the number of bandwidth values taken.
2. Choose the number of γ samples from markov chain to be used in the kernel classifier. Say we select **q** samples from the posterior. It is advisable to choose the samples from the end. For example, in this case, $t_s + M - q$ to $t_s + M$.
3. Choose a suitable kernel K for the analysis. In this article, we have chosen a zero mean unit variance gaussian kernel.
4. For a given test point x do:
   Declare a result vector res_vec=zeros
   For i in $h_1, h_2, \ldots, h_f$ do:
   $$\text{if } \left(c \sum_{t=t_s+M-q}^{t_s+M} \sum_{j=1}^{n} \mathbf{K}\left(\frac{x - x^{(j)}}{h_i} \right) \geq (1-c) \sum_{t=t_s+M-q}^{t_s+M} \sum_{j=n+1}^{2n} \mathbf{K}\left(\frac{x - x^{(j)}}{h_i} \right) \right)$$
   res_vec[i]=1
   else
   res_vec[i]=0
5. The kernel-based classifier is now given by
   $$\Psi(\mathbf{x}) = \begin{cases} 1 & \text{if } \textbf{sum}\,(\text{res\_vec}) \geq \dfrac{f+1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

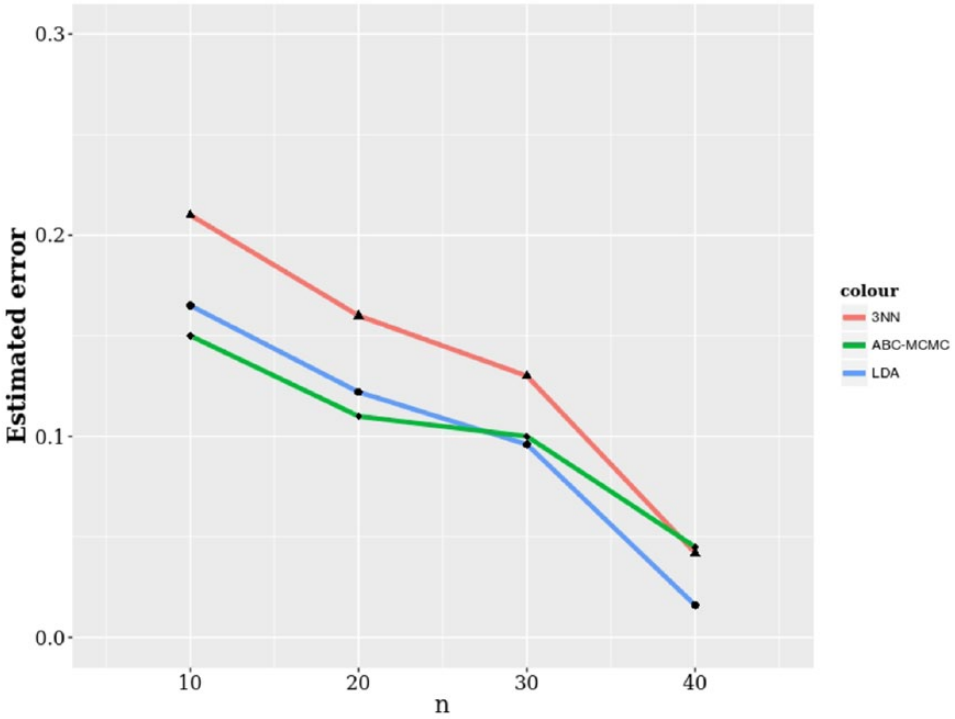

Figure 1. Average classification error rates against sample size for a fixed number of selected proteins $d = 10$. ABC-MCMC indicates approximate Bayesian computation-Markov chain Monte Carlo; LDA, linear discriminant analysis.

Numerical Experiments and Results

In this section, we demonstrate the application of the proposed ABC-MCMC classification algorithm for SRM data, using a synthetic data set generated from a subset of the human proteome. We selected a list of proteins from the Drugbank and applied tryptic digestion of proteins using the OpenMS software.[15] Because our interest is in small sample sizes, we chose simple classification rules, which are known to perform well with small samples, for comparison: LDA and k-nearest neighbor (KNN) with $k = 3$.

Synthetic SRM-MS data were generated by the model described in section "SRM-based mass spectrometry model," using the parameters in Table 1. Synthetic sample data for prior calibration were generated using the midpoint of the intervals specified in Table 1. For example, as $\phi \sim \text{Unif}(0.3,0.5)$, we take 0.4 as the initial value.

For the MCMC procedure, we consider 10 000 samples from the posterior distribution of γ. A burn-in stage of around 3000 iterations is considered. The value of prior probability was taken to be 0.5 (equally likely classes). Kernel density estimation is based on 15 MCMC samples of γ, ie, $\mathbf{q} = 15$ in Algorithm 3 (increasing this number did not show any significant difference in the results). From the initial number of 350 proteins, a t test is applied to select the top 10 to 15 proteins. We consider sample sizes $n = 10$ through $n = 40$ per class and select the number of features to be $d = 3,5,8,10$. The results displayed below are average results over 6 runs of the experiment for each combination of classification rule, sample size, and dimensionality. The classification error for each case is estimated on an independent synthetic test data set of 100 sample points.

Effect of sample size

Figure 1 displays the average error rates for the different classification rules. The number of proteins selected is fixed at $d = 10$. With the increase in sample size, we see that the total error decreases for all classification rules. An important observation is that at small sample sizes, the performance of ABC-MCMC is best, confirming the general principle of good small-sample performance by Bayesian methods.

Effect of dimensionality

The average error rates of the various classification rules against dimensionality, ie, number of selected proteins, are displayed in Figure 2, for fixed sample size $n = 10$ per class. We can observe a very strong peaking phenomenon[16]: as the number of selected proteins increases, the average classification error rates tend to go down at first, but then increase sharply, due to the small sample size, ie, small ratio between number of points over the dimensionality. One can observe that the ABC-MCMC classification rule is the most accurate one when d is large, which is in agreement with the fact that Bayesian methods tend to outperform competing techniques under small ratios of sample size to dimensionality.

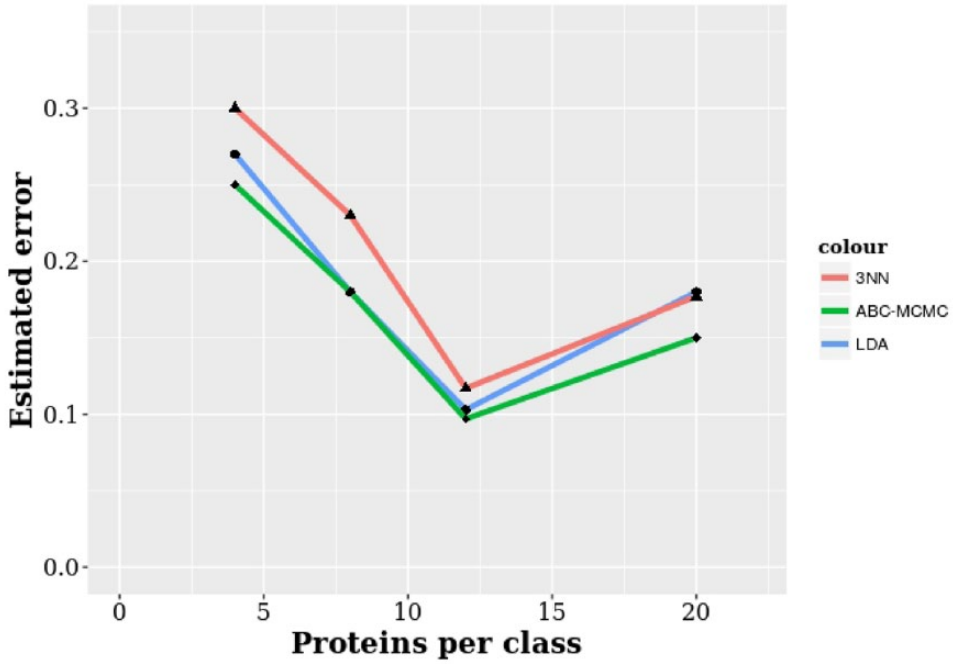

Figure 2. Average classification error rates against number of selected proteins for a fixed sample size $n = 10$. ABC-MCMC indicates approximate Bayesian computation-Markov chain Monte Carlo; LDA, linear discriminant analysis.

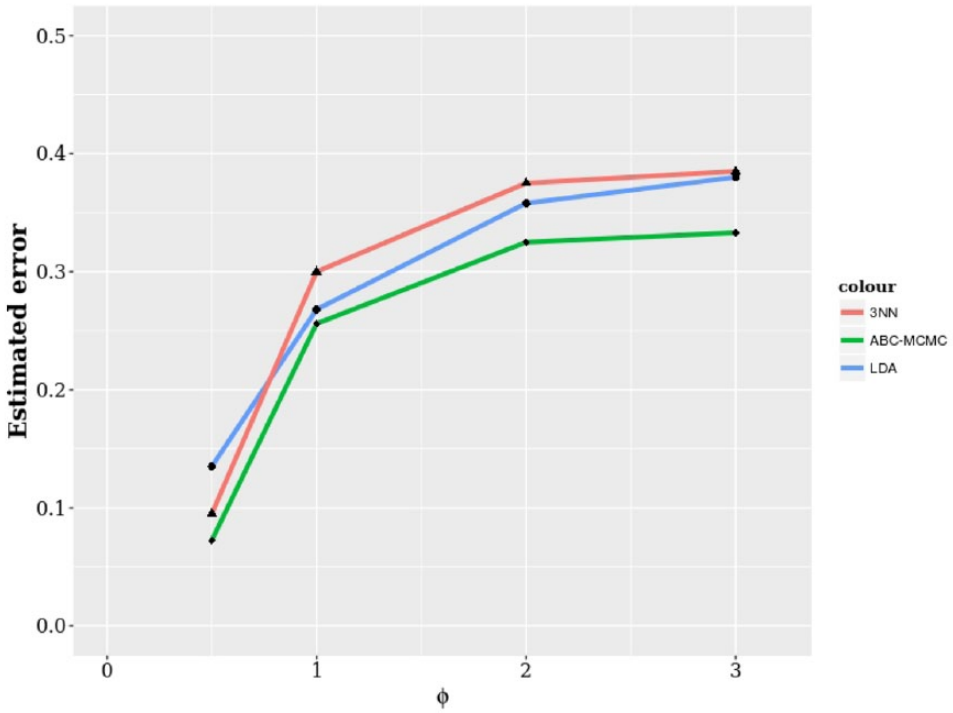

Figure 3. Average classification error rates against the coefficient of variation ϕ for a fixed sample size $n = 10$ per class and fixed number of selected proteins $d = 8$. ABC-MCMC indicates approximate Bayesian computation-Markov chain Monte Carlo; LDA, linear discriminant analysis.

Effect of variability

Here, we keep the sample size at $n = 10$ and the number of features at $d = 8$ to investigate the impact on the classification of error rate of an increasing variability of the true protein concentration values. In Figure 3, one can observe that the performance of all classification rules degrades with increasing values of the coefficient of variation ϕ; however, the performance of the ABC-MCMC algorithm is uniformly better than the others due to the small sample size $n = 10$.

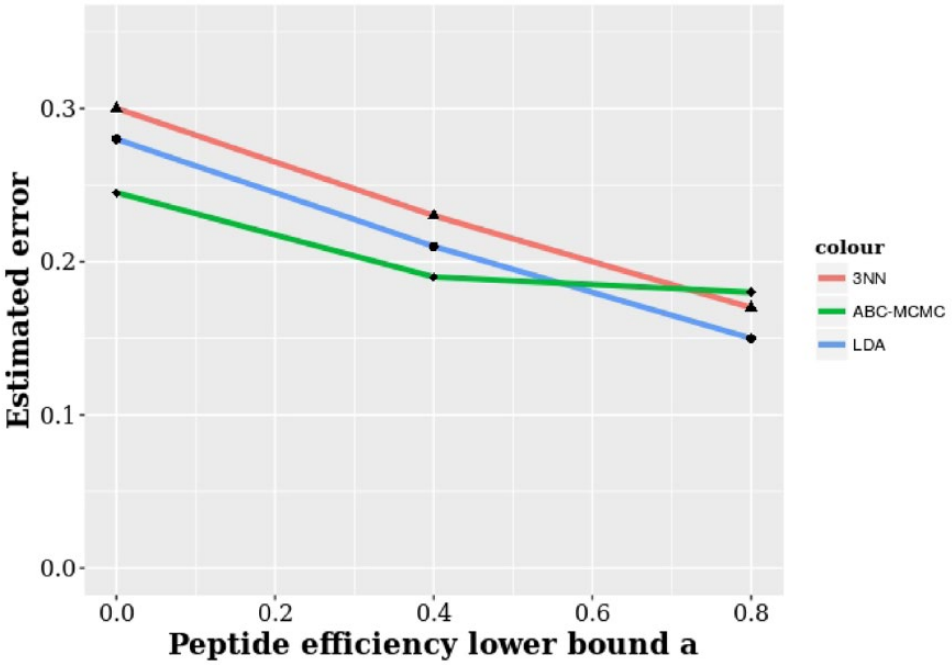

Figure 4. Average classification error rates against the lower bound for the peptide efficiency factor e_i for a fixed sample size $n = 10$ per class and fixed number of selected proteins $d = 8$. ABC-MCMC indicates approximate Bayesian computation-Markov chain Monte Carlo; LDA, linear discriminant analysis.

Effect of peptide efficiency

Finally, we investigate the impact on the classification accuracy of varying the peptide efficiency. The peptide efficiency factor *a* controls how many ions can be detected for a given peptide. Increasing this parameter uniformly increases efficiency for all peptides, which corresponds to a more accurate SRM-MS experiment. Indeed, one can observe in Figure 4 that classification accuracy tends to increase with increasing peptide efficiency. One can also observe that the ABC-MCMC classification rule displays the smallest error rates among the competing methods at low peptide efficiency, ie, in a more noisy experiment.

Conclusions

In this article, we have proposed a Bayesian approach for classifying SRM data with the goal of facilitating biomarker development. This method is a combination of ABC and MCMC. We can see that for small sample sizes, large dimensionality, or noisy data, the performance of the proposed Bayesian classifier is superior to that of other approaches. Our results are based on a subset of the human proteome selected from the Drugbank, which are submitted to tryptic digestion in silico. In addition, the prior hyperparameters are calibrated using the available data. This makes the the approach realistic and broadly applicable. Because we are studying the effects of the various parameters of the SRM pipeline on the classification error, there is a need to use synthetic data from a generative model. The results are, however, expected to be reproducible on clinical SRM data.

Author Contributions

KN and UBN conceived and designed the experiments. KN analyzed the data. KN wrote the first draft of the manuscript. UBN contributed to the writing of the manuscript. Both authors read and approved the final manuscript.

REFERENCES

1. Aebersold R, Mann M. Mass spectrometry-based proteomics. *Nature*. 2003;422:198–207.
2. Rifai N, Gillette M, Carr S. Protein biomarker discovery and validation: the long and uncertain path to clinical utility. *Nat Biotechnol*. 2006;24:971–983.
3. Hüttenhain R, Malmstrm J, Picotti P, Aebersold R. Perspectives of targeted mass spectrometry for protein biomarker verification. *Curr Opin Chem Biol*. 2009;13:518–525.
4. Ye X, Blonder J, Veenstra T. Targeted proteomics for validation of biomarkers in clinical samples. *Brief Funct Genomic Proteomic*. 2009;8:126–135.
5. Atashpaz-Gargari E, Braga-Neto U, Dougherty E. Modeling and systematic analysis of biomarker validation using selected reaction monitoring. *EURASIP J Bioinform Syst Biol*. 2014;2014:17.
6. Banerjee U, Braga-Neto U. Bayesian ABC-MCMC classification of liquid chromatography-mass spectrometry data. *Cancer Inform*. 2017;14:175–182.
7. Turner B, Zandt IV. A tutorial on approximate Bayesian computation. *J Mathemat Psychol*. 2012;56:69–85.
8. Csilléry K, Blum M, Gaggiotti O, François O. Approximate Bayesian Computation (ABC) in practice. *Trends Ecol Evol*. 2003;25:410–418.
9. Sisson S, Fan Y. Likelihood-free Markov chain Monte Carlo. In: Brooks S, Gelman A, Jones G, Meng XL, eds. *Handbook of Markov Chain Monte Carlo*. New York, NY: Chapman & Hall; CRC Press; 2010.
10. Geyer CJ. Practical Markov chain Monte Carlo. *Statis Sci*. 1992;7:473–483.
11. Wegmann D, Leuenberger C, Excoffier L. Efficient approximate Bayesian computation coupled with Markov chain Monte Carlo without likelihood. *Genetics*. 2009;182:1207–1218.
12. Dalton L, Dougherty E. Optimal classifiers with minimum expected error within a Bayesian framework part I: discrete and Gaussian models. *Pattern Recogn*. 2013;46:1301–1314.
13. Webb A. *Statistical Pattern Recognition*. 2nd ed. New York, NY: John Wiley & Sons; 2002.
14. Taniguchi Y, Choi P, Li G, et al. Quantifying E. coli proteome and transcriptome with single-molecule sensitivity in single cells. *Science*. 2010;329:533–538.
15. Sturm M, Bertsch A, Gröpl C, et al. OpenMS—an open-source software framework for mass spectrometry. *BMC Bioinform*. 2008;9:163.
16. Hughes G. On the mean accuracy of statistical pattern recognizers. *IEEE Trans Informat Theory*. 1968;14:55–63.