Bayesian Methods for Protein Quantification in Mass Spectrometry Proteomics

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

Alexander Mark Phillips

September 2019

Acknowledgements

This work was supported by EPSRC.

I would like to thank my supervisors, Professor Simon Maskell of the University of Liverpool, Professor Andrew Dowsey of the University of Bristol and Professor Andrew Jones of the University of Liverpool. Without their teaching and guidance this thesis would not have been possible.

I would also like to thank Dr Richard Unwin of the University of Manchester for providing the spike-in data sets and Alzheimer's study data used in this thesis.

My thanks also goes to Joan Gladwyn who proof-read the thesis.

I would like to thank the examiners, Professor Conrad Bessant of Queen Mary University of London and Professor Danushka Bollegala of the University of Liverpool. This version of the thesis is corrected in response to their comments following the viva.

I would like to thank Dr Ranjeet Bhamber, Dr Andris Jankevics and Dr Hanqing Liao for their support. My thanks also goes to all of the members, past and present, of the Signal Processing Group in the Department of Electronics and Electrical Engineering at the University of Liverpool for their support.

A special thanks goes to my family for their love, support and encouragement. Thank you also to my friends for being so supportive. Thank you to friends at CPC Music. And a special thank you to friends at Beatlife for providing some much-needed distraction and stress relief. Finally, I wish to thank Lizzie for her endless patience, support and encouragement.

Abstract — Bayesian Methods for Protein Quantification in Mass Spectrometry Proteomics

Alexander Mark Phillips

Current workflows in mass spectrometry proteomics are able to identify thousands of proteins in a single biological sample by breaking them down through enzymatic digestion into smaller molecules, peptides. Quantification of the differences in abundances of proteins between populations is based on measurements of these peptides at multiple charge states, features. Subsequent determination of significant changes in the abundance of proteins between those populations remains a challenge. This is complicated further by the presence of shared peptides arising from multiple proteins. This PhD thesis explores some of the statistical modelling challenges associated with the quantification of proteins in mass spectrometry proteomics. Firstly, an overview of the existing literature is presented, focusing on current methods for quantifying proteins and differential expression analysis. A current pipeline for quantifying proteins from feature-level data using Markov chain Monte Carlo sampling is then evaluated. A method by which sampling might be made more efficient by exploiting conjugate distributions is then proposed. Similar conjugate analysis is then applied to the specific problem of differential expression analysis. The analytic nature of the resulting model is exploited to achieve fast inference without the need for computationally expensive numerical integration. This enables a model comparison approach to statistical testing, with the aim of achieving calibrated estimates of false discovery rate. A Bayesian hierarchical model is then proposed for the analysis of shared peptides, with the aim of performing absolute quantification of proteoforms. Finally, overarching conclusions are drawn and recommendations for future research are discussed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A number of challenges currently exist for researchers wishing to make sound statistical inference based on the observation of thousands of proteins in a mass spectrometry proteomics experiment, which often involves very few samples and replicates. This PhD thesis explores some of those challenges and proposes approaches with which they can be tackled.

Chapter 2 provides a review of the current literature, both for proteomics and Bayesian methodologies. The depth and breadth of the current literature for proteomics and Bayesian statistics are both vast, hence Chapter 2 seeks to give a broad overview of a bottom-up analysis pipeline before focusing on the specific areas of protein quantification and false discovery rate estimation. Additionally, an overview of current methods for Bayesian analysis is presented, with a particular focus on methods for the estimation of Bayes factors.

In Chapter 3, we look at a recent development in the field, the BayesProt[1][2] analysis pipeline. BayesProt calculates sample-level protein quantifications from numerous types of mass spectrometry data, going from e.g. iTRAQ spectra measurements and performing estimation of protein quantifications, weighting each peptide's contribution to the final protein quantification by their inferred variance. Coupled with modelling of separate biological variance within test groups, BayesProt gives robust estimation of protein quantifications and importantly also provides measures of uncertainty in these protein quantification estimates, through confidence intervals, standard errors or samples from the posterior distribution. Comparing with existing methods, we observe that propagation of uncertainty is essential to robust inference of protein quantifications. This robust Bayesian estimation however comes at greater computational expense than simpler methods.

In Chapter 4, we consider some mathematical "tricks" which can be employed to effectively reduce the amount of computation required to achieve accurate Bayesian inference. We demonstrate how the fitting of a Poisson regression with MCMC can be sped-up through the use of conjugate prior distributions to reduce the size of the parameter space being sampled from.

In Chapter 5, we take this exploitation of conjugacy further, achieving totally analytic Bayesian inference of a model for differential expression testing, gaining significant reductions in computational cost with relatively minor alterations to the model used. This analytic inference concurrently allows us to employ Bayesian model comparison via computing exact marginal likelihoods to perform statistical testing and achieve calibration of the quantitative false discovery rate.

In Chapter 6, we consider a more advanced version of the BayesProt model which incorporates information from shared peptides into the protein quantification estimates. We demonstrates that effective, fully Bayesian analysis of shared peptides can be achieved through non-linear modelling, giving improved confidence of protein quantification estimates as well as providing estimates of relative abundance between proteoforms.

Finally, Chapter 7 provides a high-level overview of the conclusions reached over the course of this work, before making a number of recommendations for ways in which future work may seek to build upon it.

1.1 Contributions

A part of Chapter 2 has been previously published as part of a book chapter (Section 7.5 in [3]) which I contributed to.

The BayesProt R package described in Chapter 3 was developed prior to and during the course of my PhD by my supervisor Professor Andrew Dowsey. The validation of the software was my own work. This validation is planned to be included as part of a joint first author journal paper with Dr Ranjeet Bhamber, University of Bristol, due to be submitted for publication to Nature Biotech in June 2020.

The collapsed sampler in Chapter 4 was motivated by a discussion with my supervisor Professor Simon Maskell as a possible avenue for speeding up the BayesProt model. The derivation of the necessary conjugate distributions, coding of Stan models and evaluation of the models is my own work.

The Bayesian model comparison approach in Chapter 5 was also motivated by a discussion with my supervisor Professor Simon Maskell. The subsequent development

of the analytic model and R code is my own work. This work is planned to be submitted as part of a journal paper, due to be submitted for publication to Bioinformatics in September 2020.

The protein quantification data analysis in [2] was conducted by myself using an earlier version of the BayesProt software. The additional analysis in that journal article treating the amyloid precursor protein and amyloid beta as separate proteins was requested by Dr Richard Unwin of the University of Manchester. This exploratory analysis partly motivated the development of the shared peptide model in Chapter 6. The model and Stan code were developed by myself. The analysis treating the amyloid precursor protein and amyloid beta separately was replicated for this thesis using a model in the Stan programming language.

Conferences

Work related to this thesis was presented at a number of conferences.

ProteoMMX 2016 — poster and oral presentation

"Robust protein-level differential analysis and study-level diagnostics through Bayesian modelling" — an oral presentation and accompanying poster were presented, detailing the BayesProt model and work correcting for iTRAQ fold-change suppression through background subtraction.

HUPO 2017 — poster presentation

"Robust iTraq and TMT protein-level quantification and statistical analysis with a Bayesian mixed-effects model" — a poster presentation communicated the proteomics analysis conducted in [2], including the modelling of amyloid precursor protein and amyloid beta.

BSPR 2018 and ISMB 2018 — poster and oral presentation

"Robust iTraq and TMT proteoform-level quantification and statistical analysis through Bayesian modelling" — Although I was unable to attend the conferences, I contributed material detailing the results of the shared peptide model described in Chapter 6 on the amyloid precursor protein and amyloid beta peptide from the AD study in [2].

Chapter 2

Literature Review

2.1 Introduction

2.1.1 Quantitative Proteomics

High-throughput mass spectrometry proteomics has developed over the past two decades to allow for the identification and quantification of thousands of proteins within a biological sample[4]. Quantitative proteomics are frequently concerned with the manner in which these proteins differ in abundance between groups of samples, for instance, two groups consisting of diseased and healthy patients. Mass spectra produced from intact proteins are very challenging to interpret. Therefore, in most workflows proteins are broken down (digested) with an enzyme into shorter chains of amino acids called peptides[5]. Instead of observing proteins, the abundance of ionised peptides is measured at multiple charge states (features). Since the action of a number of different enzymes (e.g. trypsin) are well understood and relatively predictable, their action can therefore be simulated in-silico so as to deconvolute peptide to protein relationships post-digestion[6].

2.1.2 LC-MS Pipeline

In a shotgun proteomics workflow, digested samples are processed using a liquid chromatography (LC) column, out of which peptides are eluted at different points in time according to their hydrophobicity, for example. The output of the LC column is coupled to the input of a mass spectrometer (MS). This combination of techniques allows peptides to be separated along two axes, retention time (RT) and mass-charge ratio (m/z). The result is a large two-dimensional data set of "MS1" spectra across retention

time.

MS1 peaks are then selected by the mass spectrometer for subsequent fragmentation step for identification. Small m/z windows are isolated and the ions fragmented, for example, by a technique called collision-induced dissociation (CID). The fragmented peptide ion is then analysed again. The results are "MS2" spectra which are used for identification of the peptides that produced those features. In a data-dependent acquisition (DDA) workflow, a limited number of mass windows where peaks are detected are considered for MS2 fragmentation. In a data-independent acquisition (DIA) workflow, a range of m/z windows are taken for fragmentation, regardless of whether a peak is detected in that range. The m/z windows used for fragmentation in DIA are wider than those used for DDA acquisition so that the mass range can be covered within the cycle time of the instrument; MS2 spectra now contain fragmented ions from multiple peptides, complicating analysis[7].

By searching against a protein database using a search engine tool such as Mascot[8], by in-silico digestion of the proteins in that database according to the enzyme which was used for the experiment and using the RT and m/z of the precursor ion. Since the m/z and charge of the precursor ion is known, the neutral mass of the precursor can be calculated, allowing the search engine to filter the list of peptides to consider for identification based on this mass. The search space is thereby reduced in size and peptides are then identified by matching each MS2 spectrum against a theoretical spectrum generated by in-silico fragmentation of the peptide sequence.

What remains is for each of these peptide features to be quantified, the methods for which fit into one of three classes, MS1, DDA MS2 or DIA MS2 quantification.

In all of these quantification strategies, the measured intensities result from the ionisation of peptides; a given peptide sequence will ionise to a different extent compared with other peptides sequences, hence the intensities of different peptides or features cannot be compared directly[9]. Instead, only peptide or feature ratios between samples can be compared. Thus, without known concentrations of isotopically labelled standards[10] being spiked into samples, only relative quantification of peptides (and therefore proteins) between samples is possible.

2.1.3 MS1 Quantification

Label-free

In a label-free proteomics experiment, quantification can be achieved by integrating the area under the extracted ion chromatograph (XIC) across retention time[11]. For

the XIC method, under the assumption that detector response is linear, this intensity is proportional to the total ion count.

SILAC

Stable isotope labelling by amino acids in cell culture (SILAC)[12] allows differential expression analysis by growing cells in two different media rich in a particular amino acid. One medium is rich in the natural "light" amino acid and the other is rich in a heavier version of that amino acid, e.g. arginine replaced by arginine with carbon-13 atoms in place of carbon-12, or nitrogen-14 replaced with nitrogen-15. The heavy and light samples can then be processed via LC-MS together. The proteins from these cell cultures can be analysed in parallel since the heavy and light peptides from the two media appearing as pairs of peaks will be predictably separated in m/z. By comparing the areas under the two peaks, peptide ratios can be calculated. An extension of this technique allows for analysis of three cell populations.

2.1.4 DDA MS2 Quantification

iTRAQ

Isobaric tags for relative and absolute quantification (iTRAQ)[13] is a labelling method which utilises post-digestion addition of molecules containing so-called "reporter ions" to the peptides in a sample. Commercial labelling kits are available as either 4- or 8-plex, each with reporter ions of different masses. Samples are then mixed in equal ratios before being analysed by LC-MS/MS. During the fragmentation of the MS1 features, these reporter ions are cleaved from the peptide fragments and detected by the mass spectrometer. The abundance of these reporter ions in the MS2 spectra are used to obtain a quantification of that feature between the iTRAQ channels.

TMT

Tandem Mass Tag (TMT)[14][15] labelling functions in a similar fashion to iTRAQ, but with differing numbers of tags in a labelling kit (2, 6, 10 or 11 reagents). Since some of these mass tags are extremely close in mass (within 0.01 daltons), a high resolution Orbitrap instrument is required to adequately resolve between the peaks.

2.1.5 DIA MS2 Quantification

In DIA acquisition, since MS2 spectra are captured continuously with wide isolation windows, the MS2 spectra necessarily contain fragmented ions from multiple peptides; the link between MS2 spectra and specific MS1 precursors is lost[9]. Quantification must instead be based on the integration of deconvoluted MS2 peaks across retention time. Sequential window acquisition of all theoretical mass spectra (SWATH)[7] is one such DIA method gaining adoption. All available MS2 spectra are captured with these methods, in contrast to DDA methods where MS2 acquisition is triggered by MS1 signals, resulting in stochastic sampling of MS2 spectra[16]; it is for this reason that DIA acquisition shows higher reproducibility over DDA acquisition[17].

2.1.6 Poisson Noise

The detection of ions in time-of-flight and quadrupole mass spectrometers relies on the induction of a current in the detector as the ions strike it. It has been observed that these ion counts are Poisson-distributed[18][19], so it is prudent that a Poisson likelihood could be used for modelling ion counts.

However, in the case where an Orbitrap-type instrument is used for analysis, ion counts are not observed directly. The mass of ions in the trap induces an oscillating current in the detector and a Fourier transform is applied to infer the intensity of ions at that particular m/z. Since ion counts are determined indirectly, a Poisson likelihood is not a perfect model for this process, but since the intensity measured is still proportional to the true ion count, a Poisson or pseudo-Poisson noise model could still be used.

2.1.7 Chromatogram Alignment

In the cases where samples have been analysed through more than one LC-MS run, due to variability in elution time, peptides may not elute uniformly, and hence the LC-MS runs must be aligned in the retention time axis so that corresponding peptides can be matched between runs[20]. This is especially important in label-free analysis, where misalignment can result in failure to match peptides between runs, resulting in missing values.

2.1.8 Normalisation

For any of these methods it is necessary to correct for systematic differences in sample loading, amongst other factors, requiring normalisation of the observed intensities. One of the samples is arbitrarily chosen to be the reference sample, before a scaling factor is calculated for each of the other samples. To properly correct for systematic differences in sample loading, the intensities must be normalised so that differential expression can be correctly identified[21]. The techniques for this have been adopted from the analysis of microarray data in genomics[22]. Normalisation can be achieved through the use of an internal standard spiked into each sample at a known concentration, so that we can scale the intensities of the features in the other samples such that the standard intensity is consistent across all samples. However, this technique is limited by the error in quantifying the standard.

The simplest technique is that of median normalisation, which is based on the assumption that the median fold-change across an experiment is 1.0. Hence scaling factors for each sample can be calculated so as to shift the median intensity of each to match each other.

Variants on this simple median normalisation exist, most notably quantile normalisation[22] where individual quantiles (rather than just the 50% quantile, the median) of each sample are scaled to match the mean intensity of that quantile across all samples.

In MS-EmpiRe[23] Ammar et al. described a novel normalisation technique based on repeated merging of clusters of samples within a treatment group according to the similarity of peptide-level fold-change distributions. Beginning with single-member clusters each of one sample, the two most similar clusters are merged at each step by scaling the samples of one cluster by the median fold-change between them. Finally, clusters from different treatment groups are merged by scaling by the most probable fold-change between the groups.

MaxLFQ[24] implements a delayed normalisation strategy for label-free data; protein intensities are calculated with the scaling factors kept as free variables, before determining values for the normalisation by performing a non-linear least-squares optimisation procedure to minimise the total differential expression across the experiment, again based on the assumption that the majority of proteins are unchanging between samples.

EigenMS[25] uses singular value decomposition to perform batch correction; systematic bias in the residuals of the differential analysis that are not explained by the experimental design (i.e. systematic difference in protein abundance not attributable

to different treatment groups) is subtracted.

2.1.9 Methods for Protein-level Quantification

At this final stage in the pipeline we have a list of identified consensus peptide features, each with a separate quantification per sample. Through protein grouping[26], each feature will also either be annotated as a subsequence of one protein, or as a "shared" peptide that is a subsequence of more than one protein (see Figure 2.1 below).

We now wish to infer the relative abundance ratios of the parent proteins between those same samples, and either from these or directly, the higher level experimental effects (e.g. differences between treatment groups). While there are simple methods for deriving protein-level ratios between pairs of samples, more advanced statistical methods intrinsically incorporate protein-level quantitation as part of statistical differential expression analysis models that use the feature quantifications across all samples simultaneously. Due to variations in the stochasticity of peptide cleavage, in general a protein-level quantitation is more accurate when many peptide quantifications support it[27]. Statistical methods hold a key advantage here, in that that they can additionally utilise information on peptide quantification variability across samples.

We do not observe protein abundances directly, but only a perception of their abundance via a process that it is assumed we can't easily calibrate (though Chapter 6 attempts to address this). Therefore, in quantitative proteomics workflows it is standard practice to perform relative quantification of perceived abundance of proteins in different samples (as perceived via the same process). Is also therefore not standard practice to perform absolute quantification of protein abundance. Numerous methods exist to obtain (perceived) protein-level quantifications from (perceived) peptide-level data.

Current established non-statistical methods for protein-level quantification from peptide or feature-level data stretch from summing peptide abundances in order to estimate protein abundances before calculating ratios[19], to taking the median of peptide ratios[28].

MaxQuant[28] calculates protein ratios as the median of all ratios of peptides belonging to that protein. This reduces the effect of any outlying peptide ratios that could be corrupted by interferences or digestion issues, but fails to use all the available information. MaxQuant was originally developed for the analysis of SILAC data. This is extended to label-free quantification in MaxLFQ[24].

Under the assertion that ion count measurements in MS follow Poissonian statistics,

Carrillo et al.[19] investigated a number of similar schemes including averaging the peptide ratios, computing the ratio after summing the peptide quantifications in each sample, and using linear regression to compute the slope of the line fitting one sample's peptide quantifications to the other. They note the effectiveness of using the ratio after summing peptide quantifications, and a modified "total least squares" regression that minimises the orthogonal distance between the peptide ratios and line of best fit, which accounts for error in both samples' peptide quantifications. These methods are effective because errors in the relative fold-change decrease as intensity increases (as would be expected in Poissonian statistics); hence summing peptide quantifications before ratio calculation leads to proportional weighting of the more intense features. The regression approach adds a form of outlier rejection, in that intensity values are down-weighted according to their distance from the line of best fit, and therefore errors in the fold-change estimate are further reduced. The authors noted that the sum of peptide quantifications method performed marginally better, although with the added benefit of being significantly less computationally expensive than the total-least squares method.

For these simpler, non-statistical methods, an additional stage is needed to determine whether relative changes in protein abundance across treatment groups can be deemed significant[29]. Since biological variation is regularly assumed to be log-normally distributed in proteomics studies, normalised quantifications or quantification ratios are often log-transformed for statistical analysis. However, some studies have argued for the use of alternative variance-stabilising transformations[30] that account for an additional additive component approximating instrument and ion counting noise[18]. Variance stabilisation normalisation (VSN)[30] works by applying a parametric transform that removes the intensity variance's dependence on the mean, putting all samples on the same scale. However it has been demonstrated[21] that VSN will lead to the underestimation of fold-changes.

After protein-level quantifications have been obtained, downstream analysis using univariate or multivariate testing can be performed. For example, linear modelling (e.g. with Student's t-test) is often then performed to determine a subset of proteins that are likely to be differentially expressed.

Statistical Methods for Protein Quantification

Other more rigorous statistical methods which perform statistical significance testing in tandem with differential expressions estimation include the use of the mixed-effect

family of models, encompassing, for example, linear regression, multi-way analysis of variation (ANOVA) and generalised mixed-models. Often these models are fitted using techniques such as restricted maximum likelihood estimation (ReML). More sophisticated Bayesian methods for model-fitting such as Markov chain Monte Carlo (MCMC) can instead provide quantitative information with detailed credible intervals, a clear advantage over asymptotic approximation.

Missing data and outliers also present a problem. For instance, peptides can sometimes be misidentified, or if a feature is not selected for MS2 fragmentation, not identified at all. Additionally if the abundance falls below the detection limit of the mass spectrometer instrument, this missingness is not at random and therefore cannot be ignored and instead must be accounted for.

Fitting statistical models to all the feature-level data has the potential for more accurate quantification through joint inference of peptide reliability and differential quantification[31]. This comes at a cost of being more computationally intensive. These statistical models attempt to account for the inherent variability in the observed intensities due to random experimental variation, with protein digestion being a major component. The most popular modelling framework underpinning these tools is the mixed-effects model, which generalises a large cross section of statistical models including the t-test, linear regression and multi-way ANOVA.

In this framework, predictors are termed “fixed effects”. A fixed effect is one which we consider to be systematic, for example, the effect of a particular protein, peptide or condition on the observed intensity of a feature, or a batch effect between two batches. Unlike simpler models, mixed-effect models also support “random effects”. Random effects represent stochastic fluctuations that occur within larger populations and are usually represented as normal distributions with unknown variance (e.g. biological variation causing per-sample random deviations from the population mean, or batch deviations across many batches. When protein-level quantifications have already been derived, the resulting test for assessing differential expression (e.g. t-test) models biological variation as a normally distributed residual. In protein-level quantification performed by a mixed-effects model, the normally distributed residual models technical variation at the feature level instead, so it is crucial to also fit a random effect to model protein-level biological variation.

Tools such as MSstats[32] fit the mixed-effects model on a per-protein basis, employing standard methods for fitting based on ReML. Each log-transformed feature is modelled as a linear combination of peptide, condition and sample effects, with peptide and condition assigned as fixed effects and sample as a random effect (if the sample

size is adequate). MSstats also includes an optional interaction effect between peptide and condition which models feature-specific signal interferences that only appear in one condition[33]. A popular approach is to estimate a separate residual variance for each peptide, which has the effect of weighting each peptide's contribution to the protein-level quantitation by the reciprocal of its inferred residual variance[34]. However, the authors of MSstats advise that ReML will over-fit such a model, advocating that a non-linear relationship between a peptide's abundance and its variance should be enforced to avoid over-fitting[33]. This is achieved by ReML fitting of a single residual variance to all peptides, but with the variance weighted on a per-peptide basis. Through a technique called iterative reweighted least squares, the weights are initially set to unity and are then iteratively refined by rounds of locally estimated scatterplot smoothing (LOESS) curve fitting to the model residuals against predicted peptide abundance followed by ReML model refitting.

Prior to developing MSstats, Clough et al. proposed two ANOVA models for label-free data[35], one "fixed effects", and one "mixed effects":

$$y_{ijk} = \mu + \beta_i^{\text{Group}} + \beta_j^{\text{Feature}} + \beta_{ij}^{\text{Group, Feature}} + \beta_{ik}^{\text{Group, Subject}} + \epsilon_{ijk} \tag{2.1}$$

$$\sum_{i=1}^{g} \beta_i^{\text{Group}} = \sum_{j=1}^{f} \beta_j^{\text{Feature}} = \sum_{i=1}^{f} \beta_{ij}^{\text{Group, Feature}} = \sum_{i=1}^{g} \beta_{ij}^{\text{Group, Feature}} = 0 \tag{2.2}$$

$$\epsilon_{ijk} \sim N(0, \sigma^2) \tag{2.3}$$

where for feature j from a subject k in group i, y_{ijk} is the observed log-intensity. β_i^{Group} and β_j^{Feature} are fixed effects for each group and feature respectively, and $\beta_{ij}^{\text{Group, Fetaure}}$ is a fixed effect for the interaction between group and feature. ϵ_{ijk} is the residual error with variance σ^2. $\beta_{ik}^{\text{Group, Subject}}$ is the subject effect, which is has different constraints in the two models. The fixed effects ANOVA has the additional constraint:

$$\sum_{k=1}^{n} \beta_{ik}^{\text{Group, Subject}} = 0 \tag{2.4}$$

whereas the mixed effects ANOVA instead has the constraint:

$$\beta_{ik}^{\text{Group, Subject}} \sim N(0, \sigma_k^2) \tag{2.5}$$

In the fixed effects model, the subject is considered a fixed effect, whereas in the mixed effects model it is considered a random effect. The authors claim that this

distinction allows the user to choose whether they wish to treat subjects as individuals between whom they make comparisons, or as random samples from wider populations in order to compare the average protein quantifications between those populations[35].

Oberg et al.[36] (and the accompanying paper by Hill et al.[37]) describe an ANOVA model for iTRAQ quantification which takes into account sources of variability:

$$\begin{aligned} y_{ijkqsl} = \mu + (v_{ql} + b_q) + (\beta_i^{\text{Protein}} + \beta_j^{\text{Peptide}}) \\ + (\beta_{ik}^{\text{Protein, Group}} + \beta_k^{\text{Group}} + \beta_{jk}^{\text{Peptide, Group}}) + \epsilon_{ijkqsl} \end{aligned} \tag{2.6}$$

where for measurement s of peptide j from protein i, which comes from the iTRAQ channel l in experiment q, and belonging to comparison group k, y_{ijkqsl} is the observed log-abundance and μ is the model intercept, the abundance of the arbitrarily chosen reference level. The first group of terms (v_{ql} and b_q) account for systematic differences between iTRAQ channels and iTRAQ runs. An important difference between this and other similar models is that the second group of terms ($\beta_{ik}^{\text{Protein, Group}}$, β_k^{Group} and $\beta_{jk}^{\text{Peptide, Group}}$) are considered as random effects, that is, it accounts for some stochastic elements of the enzymatic digestion. The third group of terms are the ones of interest in a typical proteomics experiment, the systematic differences at peptide and protein level between comparison groups.

Jow et al.[38] present a fully Bayesian ANOVA model to perform concurrent normalisation and differential expression analysis. Drawing some inspiration from Oberg et al. [36], their model utilises Bayesian variable selection to classify differentially and non-differentially expressed proteins. The modelling of all proteins concurrently also removes the need for multiple hypotheses correction (as noted in [39]).

Goeminne et al. in MSqRob[31] presented three improvements to increase the robustness of the mixed-effects approach: Ridge regression, Huber weights and empirical Bayes. Ridge regression is adopted to reduce over-fitting by penalising the peptide effect. This essentially transforms the peptide effect from a fixed effect into a per-protein random effect. The consequence is to widen the scope of differential expression testing; rather than being based only on the peptides present, the scope now includes a population of theoretical peptides from which the quantified peptides are but part. Since estimating this population's variance can only be achieved with significant uncertainty when a protein is supported by only a few quantified peptides, statistical testing becomes more conservative in these cases. Rather than estimate per-peptide residual variances, an M-estimation approach with Huber weights is used to down-

weight individual outlier quantifications. Through a similar empirical Bayes procedure to limma[40], the residual variance estimates of proteins with few observations are made more reliable by borrowing strength from the variance estimates of other proteins in the experiment.

Sticker et al.[41] built on the above in MSqRobSum by instead fitting the model in two parts, whereby peptide quantifications are initially summarised up to protein quantifications through a reduced version of the same mixed-effects model, before continuing with differential expression analysis as before. Though they expect a reduction in performance, this simplified summarisation greatly decreases the computational complexity of the model fitting, also allowing for analysis of protein quantifications by alternative methods.

The and Käll[42] recently presented a model which incorporates the identification error probability of each identified peptide into a Bayesian graphical model, combining the normally separate concepts of error rates for identification and quantification into one. Rather than fitting the hierarchical model across all proteins, they use an empirical Bayes method to set the hyper-parameters and fit the model on a per-protein basis. They compare the their approach with a MaxQuant-based pipeline, demonstrating an increase in sensitivity and a better control of the false discovery rate.

The proteome informatics pipeline will result in many consensus features missing quantifications for one or more samples. There are two main mechanisms for this missingness: low intensity features are much more likely to be missed due to insensitivity in the feature detection method i.e. these quantifications are 'censored'; features can be missed at random, due to a combination of technical (e.g. ion-suppression effects) and informatics issues (e.g. failure to deconvolute co-eluting interferences). Ignoring all missing data will reduce the sensitivity of differential expression analysis, as protein quantifications will be over-estimated in conditions with greater numbers of censored values. Conversely, setting all missing data to zero will both over-estimate differential expression and bias quantitation where the missingness is completely at random.

Karpievitch et al.[34] have presented a mixed-effects model that can compensate for these missingness mechanisms, and optional impute the missing data. Given a study-wide heuristic estimate of the probability a missing quantification is at random, their model estimates peptide-specific censoring thresholds and hence the distribution of intensity values each missing quantification could have represented.

QPROT[43] provides a framework for testing for differential expression of protein-level quantifications from label-free data. It performs differential expression testing directly on protein-level quantifications which have been determined by other means

(e.g. MaxQuant). The intensity data y_{ij} of protein i from sample j is assumed to be log-normally distributed:

$$\log(y_{i,j}) \sim N(\mu_i + d_i T_j, \sigma_i^2) \tag{2.7}$$

where u_i is the mean intensity of protein in the group 0 (e.g. control group), d_i is the log-fold-change between the two groups 0 and 1 (e.g. the diseased group) and T_j is a binary indicator of which group a sample belongs to. QPROT handles missing data in a similar fashion to Karpievitch et al.[34] by modelling the probability of a missing datapoint being missing at random and the distributions of probable missing values. The model is fit using MCMC, before the samples for each d_i are used to calculate a Z-statistic for each protein. Finally it provides mixture model-based FDR estimation. The authors demonstrate that QPROT shows improved sensitivity over limma[44], an empirical Bayes procedure originally developed for gene-expression studies but occasionally employed with some success in quantitative proteomics studies[45].

The final result of the presented quantitation pipeline is a list of proteins which we have identified as being differentially expressed according to our experimental design, along with a measure of our confidence in this assertion, the FDR, and a measure of how much we believe them to have changed i.e. their ratios or fold changes. These results could then be used to present a set of proteins for further analysis, whether by pathway analysis or as candidates for developing a biomarker prediction panel.

2.2 Shared Peptides

Multiple molecules dubbed "proteoforms" with similar amino acid sequences can ultimately be produced by the same gene[46]. When subjected to enzymatic digestion, the same peptide can be produced by multiple proteoforms. These peptides are said to be "shared" between the parent proteoforms. The differences between these proteoforms can for example be as a result of: genetic differences where different alleles of the same gene lead to differences in protein sequences amongst individuals in a population; differences in transcription of the gene to RNA due to alternative splicing; or due to post-translational modifications, for example phosphorylation, where the addition of a phosphoryl group to an amino acid changes a protein's function. Shared peptides are also produced when different genes within a gene family give rise to very similar protein sequences. These shared peptides present a problem for the peptide-to-protein

quantification stage of a proteomics pipeline since using relative peptide abundance as a proxy for relative protein abundance is only viable in cases where the peptides are unique to that protein[27] and are therefore often discarded. In a typical protein database, shared peptides can account for as much as 50% of the data[47], data which if handled properly can result in more accurate estimates of differential expression[27][48]. Furthermore, in the cases where a protein has no unique peptide, the removal of shared peptides means that the protein's presence or corresponding quantification cannot be inferred.

For example, Figure 2.1 considers the case where proteins 1 and 2 are represented in results by peptides A, B and C, and B, C and D respectively. Ignoring shared peptides B and C we can infer the presence of both proteins 1 and 2 by the presence of peptides A and D. Then suppose that protein 3 is typified by peptides B and C only; ignoring shared peptides would mean that protein 3 is undetected.

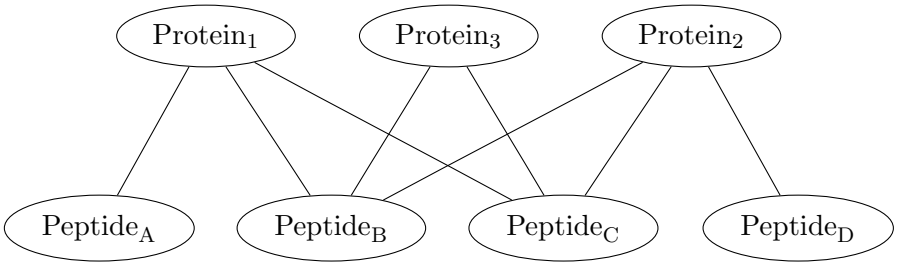

Figure 2.1: Protein–Peptide Relationship Example — Proteins 1, 2, 3; peptides A, B, C, D. Proteins 1 and 2 are represented in results by peptides A, B and C, and B, C and D respectively. Protein 3 is represented by peptides B and C. In the event that shared peptides B and C are discarded prior to analysis, the presence of proteins 1 and 2 can be inferred, via peptides A and D. However, in this situation, we would be totally unable to infer the presence of protein 3.

2.2.1 Statistical Models Accounting for Shared Peptides

The Bayesian model described by Webb-Robertson et al.[49] identifies and quantifies proteoforms by comparing the quantification patterns of individual peptides and collecting peptides with similar quantification patterns together to represent individual proteoforms. Maximisation of the posterior probability is then used to infer the most likely combination of proteoforms.

Jin et al.[50] present a methodology that is restricted to peptides derived from proteins determined to be biologically related. Combining peptide ratios to determine

the common abundance of the proteins in the group, the authors suggest that shared peptides be included only in the cases where the biologically related proteins do not show different differential expression relative to the control group.

A model for handling of shared peptides in the context of spectral counts is described in [51]. Spectral counts of shared peptides are distributed between parent proteins in accordance with the ratios of the total unique spectral counts for each of the parents. The authors compare this strategy to that where shared peptides are excluded and where shared peptides counts are simply assigned multiple times to each parent without ratio-ing, summing the signal of all peptides assigned to a protein. They demonstrate that the distribution of the counts shows better reproducibility over using only unique peptides and summation of all peptides. The same technique is applied to label-free protein quantification in [52] with similar conclusions.

Blein-Nicolas et al.[48] proposed a non-linear model where the measured log-transformed quantification of a peptide, $\log(I_{itr})$, is equal to the sum of: the log-sum of the quantifications of the proteins it is a part of, a peptide random effect $\epsilon_i^{\text{Peptide}}$, random error due to biological variation B_r, random error due to technical variation C_{tr}, and the residual error ϵ_{itr}:

$$\log(I_{itr}) = \log(\sum_k \delta_{ik} \exp(\theta_{kt})) + \epsilon_i^{\text{Peptide}} + B_r + C_{tr} + \epsilon_{itr} \tag{2.8}$$

Crucially, this model is evaluated across all proteins at once, and uses all the available information to calculate the protein quantifications, the authors demonstrating that this improves the accuracy of the estimations of parameters compared to a model quantifying one protein at a time. This comes at significant computational expense, however, as Bayesian MCMC is utilised for inference.

The model of Dost et al.[27] represents the peptide-protein relationships as a bipartite graph (similar to figure 2.1), where an edge between a protein and a peptide indicate that the protein is a parent of that peptide. Crucially, peptides can have more than one parent protein, and proteins more than one constituent peptide. Peptide ratios between groups are then calculated as the ratio of iTRAQ reporter tag intensities, before using a linear programming approach to fit the protein ratios optimally. Importantly, since this method relies on the *ratios* between assays, the ionisation of each peptide is unimportant since it is effectively cancelled out.

Gerster et al. present SCAMPI[53], a framework for protein quantification which includes shared peptides in its analysis to infer protein abundances. Starting from

peptide-level abundance estimates, the model makes a linear approximation by the assumption that peptide log-intensity is proportional to the sum of protein log-intensities, using maximum likelihood estimation or a least squares estimation to derive point estimates for the parameters. The method however does not apply to relative quantification studies (as noted in the supplementary material of [54]).

Jacob et al.[54] present the PEPA test software which allows for the inclusion of shared peptides. As in [53], the authors make a linear approximation to the peptide intensities relationship to protein intensities and use both MLE and MAP estimators to fit the model. Showing results on data sets with shared peptides added in-silico, PEPA is demonstrated to achieve increased recall over other methods[31], especially on data sets where the proportion of artificial shared peptides was high.

He et al.[55] describe a method for dealing with shared peptides in a label-free context. They suggest that the MS1 signal for a protein i can be calculated as the sum of MS1 signals from it's unique peptides: Further to this they distribute the MS1 signal from shared peptides according to the ratios of MS1 signals of the associated unique peptides of the parent proteins. However, this fails to account for differences in detectability that may be experienced by individual peptides. Hence incorrect conclusions may be drawn in cases where the peptides of one protein in the protein group are say, poorly ionised, leading to a smaller proportion of the shared peptide being assigned to that protein.

With the exception of [27], which is based on ratios between iTRAQ tags intensities, it should be noted that the above models for shared peptide analysis do not take into consideration the fact that the detectability of each peptide observed in an analysis is independent and therefore the observed intensity of a peptide is not necessarily representative of the true underlying abundance of the parent protein.

2.2.2 Null Hypothesis Testing

Student's and Welch's T-Tests

For case-control experimental design, in many experimental workflows it is common for a Student's t-test to be applied to logged protein abundances in order to determine the statistical significance of the difference in abundances.

Note that for clinical studies, Welch's t-test should be considered, since we cannot assume that the two cohorts have the same population variance.

In either case, we are then left with a p-value for each protein, the probability that the observed data or more extreme data would occur if the null hypothesis (that is,

that there is no difference in abundance) were true.

Limma

The limma R package[40], though originally developed for the analysis of microarray data in genomics studies, has also been used to perform protein differential expression testing in proteomics[56]. It uses a linear model to calculate effect sizes combined with an empirical Bayes procedure on the residual variance which shrinks the estimates of variance towards a pooled estimate, thereby borrowing strength across proteins. It subsequently uses the per-protein estimates of effect size and variance to calculate a moderated t-statistic for each protein. The empirical Bayes procedure has been updated since the original publication to optionally allow for the down-weighting of outliers in the estimation of the hyperparameters[57].

2.2.3 Multiple Testing Correction

Originally applied to the protein identification problem[58] where the use of target decoy search databases allows for identification of an appropriate null distribution, the need for multiple testing correction has since been recognised[59] in quantitative proteomics, becoming a necessary part of the pipeline.

In a quantitative proteomics discovery experiment, the goal of the study is centred around identifying a set of candidate proteins whose behaviour between treatment groups is significant, such that further, targeted studies can be carried out, studying the behaviour of these proteins in greater detail. Selecting a significant set requires statistical testing to be carried out to discern those proteins which are likely to be differentially expressed. When performing multiple statistical tests, whether through Student's or Welch's t-tests or through the use of a more complex model, it is accepted that there will be a number of false positives in the set that has been declared significant. When making many null hypothesis tests, focus shifts from determining which tests are statistically significant at some predetermined level (say $p < 0.05$), towards population-level metrics of the number of false positives.

Bonferroni

Controlling the Family-wise Error Rate (FWER) is the simplest multiple testing correction; controlling the probability of making at least one type-I error, that is, the probability that at least one of the discoveries in the declared set is a false positive. This can be achieved by applying Bonferroni correction to the sorted list of outputted

p-values. This same quantity can also be calculated for Bayesian methods outputting posterior error probabilities (PEPs) in a straightforward manner:

$$FWER_n = \min\Big(1.0, \sum_{i=1}^{n} PEP_i\Big) \tag{2.9}$$

Control of the FWER is often conservative; it comes at the expense of a reduction in statistical power[60][61].

False Discovery Rate

Since we are testing multiple proteins (multiple hypotheses), it is essential that the p-values are then adjusted to control the False Discovery Rate (FDR), which is the expected proportion of false positives in the set that we declare to be significant.

In many cases it is preferable to instead control the False Discovery Rate, that is the probable proportion of Type-I errors in the significant set, the *proportion* of false positives in the significant set.

Rather than attempting to eliminate these false positives, instead bioinformaticians in proteomics attempt to limit the number of false positives in the significant set by controlling the false discovery rate (FDR) to be below some specified level, most often 5% of the set.

The concept of calculating quantitative FDR and techniques for multiple testing correction have been borrowed from the field of genomics, where it is not uncommon for microarray experiments to contain tens of thousands of probes[62]. In contrast, a typical proteomics experiment might only identify a few thousand proteins[63]. Combine this with the small sample sizes typical of a protoeomics experiment, and the result can be that very few proteins are declared as discoveries[64].

Benjamini–Hochberg

In the paper[60] which introduced the concept of a False Discovery Rate, Benjamini and Hochberg describe a procedure, now known as the Benjamini–Hochberg (BH) procedure, for controlling the FDR given a set of p-values generated from t-tests.

Storey's q-value

Storey's method[65] for FDR estimation attempts to improve on the BH procedure by estimating the proportion of nulls in the set π_0 so as to better control the FDR, also

demonstrating an increase in power over Benjamini–Hochberg[61]. It is implemented in the qvalue Bioconductor R package. The Benjamini–Hochberg procedure is equivalent to the Storey method with a fixed value of $\pi_0 = 1.0$.

It has been previously demonstrated[31] that the above methods and many others tend to underestimate the FDR when compared against the empirical false discovery proportion (FDP), which is calculable when analysing simulated or spike-in data. This is undesirable since it is expected that FDR values are conservative[59]. However, these FDR methods are reliant on p-values being calibrated (accurate), hence it could be the case that the prior modelling is at fault, rather than the FDR estimation.

Furthermore, both BH and Storey's method make the assumption that all tests being considered are exchangeable, i.e. in each test the probability of a discovery is equally likely[61]. This is not always the case; for instance proteins may appear in differing numbers of samples in the study, meaning that the tests have differing sample sizes and hence sampling error.

Extensions to BH and Storey's q-value

Korthauer[61] highlights two counterparts to the above methods which incorporate additional information to estimate the false discovery rate, counteracting the violation of the assumption of exchangeability. IHW[66] serves as an analogue of the Benjamini–Hochberg, using a modified version of the BH procedure to weight p-values. The weights are informed by domain-specific values; for instance the authors analyse a quantitative MS proteomics experiment, using the total number of peptides for each protein to calculate the weighted p-values. Similarly, the method of Boca and Leek[67] is analogous to Storey's q-value, and takes a similar approach to [66] but goes further by approximating a π_0 value individually for each test informed by the covariates. For both these methods, [61] notes that each reduces to its respective counterpart with the use of an uninformative covariate.

Bayesian Methods

For Bayesian techniques which output posterior error probabilities (PEP), after sorting the list of proteins by ascending PEP, at each point along the length of the list the false discovery rate of the preceding set can be calculated as the cumulative average of PEP[68][69]:

$$FDR_n = \frac{1}{n}\sum_{i=1}^{n} PEP_i \tag{2.10}$$

The ash[70] package for the R programming language for adaptive shrinkage applies a different empirical Bayes approach to limma[40]. Making the assumption that effects are unimodal about zero, ash estimates a better prior on fold-change and is able to shrink effect estimates towards the mode and provide estimates of false discovery rate. Additionally, like [66] and [67], ash incorporates additional information along with observed effect sizes to improve the power of the test. Specifically, ash takes standard errors of the effect estimates. The author also recommends the use of the "local false sign rate" (equivalent to a type-S error) as a test probability and an analogue to the Storey q-value, the s-value (calculated in the same fashion as equation 2.10 above), which they claim to be more robust than their reported q-values.

2.3 Bayesian Statistics

Bayesian inference offers an alternative to the frequentist school of statistics, based around the central concept of a prior distribution. The parameters of a model are assigned a prior distribution, conceptualising prior knowledge regarding the likely values of parameters. This prior distribution is then updated using Bayes' rule conditional on the observed data to give the posterior distribution.

2.3.1 Bayes' Rule

$$P(\theta|y) = \frac{P(y|\theta)\cdot P(\theta)}{P(y)} \tag{2.11}$$

$$\propto P(y|\theta)\cdot P(\theta) \tag{2.12}$$

The general goal of Bayesian inference is to make calculations to determine the form of the full joint posterior density. In the case in which the parameters of interest θ_j are some subset of $\theta_{1,\ldots,J}$, we instead summarise the full posterior density by integrating across all other parameters to determine the marginal posterior density of θ_j:

$$P(\theta_j|y) = \int_{\theta_{1,\ldots,j-1,j+1,\ldots,J}} P(\theta|y)d\theta_{1,\ldots,j-1,j+1,\ldots,J} \tag{2.13}$$

The denominator of (2.11) is known as the marginal likelihood or model evidence. In general, this quantity takes the form of an intractable integral, and therefore, we most often use the alternate formulation for the unnormalised posterior density (2.12).

2.3.2 Bayesian Analysis Workflow

The process of performing a Bayesian data analysis can be deconstructed into three parts: the construction of the statistical model and the selection of priors, the estimation of the model conditional on the observed data, and the subsequent evaluation of the model estimate.

The parameters of interest in any statistical model are normally unobserved by the data collection process; for example, in the context of a quantitative mass-spectrometry proteomics experiment the parameters of interest might include the mean log-fold-change between two treatment groups, δ, while the observed data $\mathbf{y}$ consists of ion counts from a number of different peptide ions in multiple samples. The (Bayesian) model defines the supposed relationship between the variables of interest, any other latent variables (such as variance parameters) and the observed data. Where appropriate, prior distributions should be chosen for the latent variables in order to reflect the prior knowledge regarding the likely values of these parameters. This prior knowledge ideally encapsulates the expert knowledge of the experiment and the process by which the data was collected. For example, we might decide that a Poisson distribution is a suitable distribution to describe the likelihood of observing a number of peptide ions, where the mean rate for a peptide i is conditional on some function of δ and any number of other observed and unobserved variables, $\boldsymbol{\theta}$:

$$P(y_i|\boldsymbol{\theta}, \delta) = \text{Poisson}\left(f_i(\boldsymbol{\theta}, \delta)\right) \tag{2.14}$$

We might also believe that the prior distribution of likely values for the log-fold-change between treatment groups is well-described by a normal distribution with some mean, μ, and standard deviation, σ:

$$P(\delta) = \text{Normal}(\mu = 0, \sigma = 50) \tag{2.15}$$

In Chapter 6, Section 6.2 serves as an example of this process of determining a generative model, establishing the relationship between the observed data and the parameters of interest in the context of a hierarchical Bayesian model for the analysis of shared peptide data.

Determining the posterior distribution conditional on the observed data analytically often requires evaluation of an intractable integral for all but the simplest of models. Instead, estimation of the posterior distribution is often performed using algorithms such as Markov chain Monte Carlo simulation (see Section 2.3.4 below), which provide approximations of the posterior density.

Finally, the resulting posterior estimates should be evaluated in the context of the research question. Most often this involves calculating summary statistics and plotting marginal posteriors for each of the parameters. In our quantitative proteomics example, we might be interested in the probability that the mean log-fold-change between treatment groups is greater than zero, $P(\delta > 0|\mathbf{y})$.

Further exploration of this process of Bayesian data analysis is described in numerous texts, most notably in [71].

2.3.3 Conjugate Priors

It is possible to make a choice of prior distribution such that, given a particular likelihood distribution, the resulting posterior is of the same form as the prior distribution[72]. For example, for a normal likelihood with known variance, a conjugate prior for the mean μ is itself a normal distribution. Similarly, for a Poisson likelihood, a conjugate prior for the rate is the gamma distribution[71] In this instance it is possible to perform analytic updating of the prior distribution conditional on observed data, without the need for numerical integration.

2.3.4 Markov Chain Monte Carlo Methods

The application of Bayesian statistics to real-world problems was limited up until the point at which advances in computation enabled the development of algorithms such as Markov chain Monte Carlo (MCMC) methods, allowing statisticians to fit more complex models to larger amounts of data.

At their most basic level, MCMC methods generate representative samples from a target probability density, $\pi(x)$ by making random walks through that distribution. This is done by proposing new samples with a proposal distribution $q(x'|x)$ and randomly accepting the proposed jump with probability $a(x'|x)$.

This acceptance probability is derived by assuming that the sampler has progressed to the point where the probability of starting at a point x is the same as the target density $\pi(x)$ and that the probability of starting at a point x and moving to a point x' is equal to the probability of starting at x' and moving to x:

$$\pi(x)q(x'|x)a(x'|x) = \pi(x')q(x|x')a(x|x') \tag{2.16}$$

which can be arranged to give:

$$\frac{a(x'|x)}{a(x|x')} = \frac{\pi(x')}{\pi(x)}\frac{q(x|x')}{q(x'|x)} \tag{2.17}$$

Both acceptance probabilities are at most 1.0, and it is desirable to accept the maximum number of proposed jumps as possible. Hence:

$$1 = \max(a(x'|x), a(x|x')) \tag{2.18}$$

which can be rearranged to give:

$$1 = \min\left(\frac{1}{a(x'|x)}, \frac{1}{a(x|x')}\right) \tag{2.19}$$

$$\implies a(x'|x) = \min\left(1, \frac{a(x'|x)}{a(x|x')}\right) \tag{2.20}$$

Finally, substituting in (2.17):

$$a(x'|x) = \min\left(1, \frac{\pi(x')}{\pi(x)}\frac{q(x|x')}{q(x'|x)}\right) \tag{2.21}$$

The path that a sampler takes through a distribution forms a Markov chain, which is constructed in such a way that the stationary distribution of the Markov chain is the target posterior distribution, $\pi(x)$.

MCMC methods generally vary in terms of how these proposal distributions are constructed; from simple normal distributions centred on the current position, to more complex proposals based on physical simulation[73].

Metropolis and Metropolis–Hastings

Metropolis et al.[74] first described an algorithm for sampling from a probability distribution, using a simple random walk. Metropolis assumes that proposal distributions are symmetric (i.e. that $q(x|x') = q(x'|x)$), the result of which is that (2.21) is reduced to:

$$a(x'|x) = \min\left(1, \frac{\pi(x')}{\pi(x)}\right) \tag{2.22}$$

such that proposed jumps into regions of higher density are always accepted ($\frac{\pi(x')}{\pi(x)} > 1$).

Hastings[75] generalised this algorithm to apply to non-symmetric proposal distributions to give the Metropolis–Hastings algorithm. Gibbs Samplers and Hamiltonian Monte Carlo samplers (see below) act as special-cases of this algorithm.

The Metropolis and Metropolis–Hastings algorithms each require that the proposal distribution be tuned so that it is suited to the target distribution: jumps that are too small mean that convergence is slow; jumps that are too large mean that a high proportion of jumps will be rejected by the accept–reject procedure[76]. By monitoring the acceptance ratio and adjusting the scale of the proposal distribution accordingly, an adaptive proposal distribution can be used, leading to more efficient sampling over a fixed proposal[77].

Gibbs Samplers

Gibbs samplers[78] eliminate the accept-reject mechanism by constructing proposals in such a way that the acceptance probability is always 1.0. By constructing models such that the marginal posterior for each parameter conditional on all other parameters is analytic (through exploiting conjugate prior distributions), Gibbs samplers are able to traverse the parameter space in a more efficient manner than a random-walk Metropolis sampler, the result of which is higher effective sample size.

Metropolis-within-Gibbs

By performing standard Metropolis steps in between Gibbs sampling steps, more complex models for which some parameters do not have analytic marginal posteriors can still be sampled from.

Hamiltonian Monte Carlo

Gibbs samplers, despite showing improved efficiency over Metropolis–Hastings, suffer from high autocorrelation when compared with more advanced proposal schemes: the MCMC chain in a Gibbs sampler can only move in directions perpendicular to the axes of the parameter, effectively only allowing changes to one parameter at a time; in highly constrained regions of parameter space this can lead to undesirable and inefficient random walk behaviour[71].

Hamiltonian Monte Carlo (HMC)[73] solves this problem by allowing the chain to make jumps in directions non-perpendicular to the parameter axes. Like MCMC methods in general, HMC first was developed for applications in physics, before later being used for tackling more general statistical modelling problems. HMC utilises Hamiltonian dynamics to generate proposals from the joint posterior distribution. By considering the negative log-posterior as a potential field and simulating a particle moving through the parameter space, the particles position after a specified number of timesteps is used as the new proposal, before applying the MH ratio to determine acceptance. The result is an MCMC sampler which, after tuning of parameters, is able to efficiently traverse the parameter space, resulting in lower autocorrelation between successive samples and thus higher effective sample size[79] (see below for details of autocorrelation and effective sample size).

No-U-Turn Sampler

HMC samplers, despite their ability to sample from the parameter space more efficiently, still require careful tuning to achieve good performance[80]. The No-U-Turn sampler (NUTS)[79] seeks to eliminate the need for tuning of an HMC sampler. By adaptively tuning the step-size parameter during the initial warm-up phase of the MCMC iteration and choosing the number of steps at each iteration, the NUTS sampler is able to achieve similar (and in many cases, better) performance to a hand-tuned HMC sampler[79].

The reference implementation of the No-U-Turn Sampler is in the Stan programming language[81]. Stan provides a C++-derived model specification language, which allows users to describe a log-posterior function, either directly or by providing sampling statements which describe a model. Stan models are then cross-compiled to C++ before being compiled into executable programs.

2.3.5 Convergence Diagnostics

For any MCMC method, it is important to ensure that outputted samples are representative of the desired joint posterior distribution. Starting multiple MCMC chains at disperse initial positions helps to give confidence that the resulting samples are representative of the posterior; a single chain may find itself converging to a single mode of a multi-modal posterior.

Gelman–Rubin Convergence Diagnostic

One such method for determining convergence (or rather lack-thereof) is the Gelman–Rubin Convergence Diagnostic, $\hat{R}$[82][71]. The Gelman–Rubin Convergence Diagnostic assesses the variance of each parameter within and between MCMC chains and calculates the Potential scale reduction factor, $\hat{R}$. A large value of $\hat{R}$ implies that more MCMC iterations are required, since there exists evidence that the chains are either not stationary (having converged to the target distribution) or that they are not exploring the same region parameter space. With a value close to 1.0, (a condition such as $\hat{R} < 1.2$ is often used as a decision metric), one can be reasonably confident that the chains have converged.

It should be noted that the Gelman–Rubin Diagnostic cannot confirm convergence, it can only identify non-convergence; a value of $\hat{R}$ close to 1.0 does not guarantee that MCMC chains have converged, only that there is no evidence of non-convergence.

Additionally, the Gelman–Rubin diagnostic is only valid for parameters whose distribution is approximately Gaussian. In the cases of parameters, whose values are constrained (say, to be strictly positive or in the interval $[0, 1]$), then their MCMC samples should be transformed to be $\in \mathbb{R}$ (e.g. with a log or logit transform).

2.3.6 Effective Sample Size

One of the limitations of MCMC methods is that consecutive draws from the posterior distribution are correlated by the nature of the Markov chain property. This autocorrelation does not pose an immediate problem, however practitioners need to be aware that it does however reduce the effective amount of information contained within those MCMC samples. Therefore it is prudent to, after a sampler has been determined to have converged on the target distribution, to calculate the "Effective Sample Size" for the outputted MCMC samples[71]. From this it can be decided whether a sufficient number of MCMC iterations have been run to make inferences about parameters of interest (or summary statistics) to a desired degree of accuracy.

2.3.7 Empirical Bayes

Empirical Bayes methods seek to approximate full hierarchical models by estimating the hierarchical prior distributions from the data[83]. These methods have been successfully applied to problems in genomics[40] and proteomics[31], with a particular focus on estimation of false discover rate[70].

2.3.8 Sequential Monte Carlo Samplers

A recent advancement is that of Sequential Monte Carlo (SMC) samplers. In contrast to the population of independent Markov chains used in MCMC samplers, SMC samplers instead rely on a population of particles where information regarding the posterior is effectively distributed across the particles. SMC samplers can be readily parallelised[84], including across high-performance distributed computing systems[85].

2.3.9 Bayes Factor

The Bayes factor between two competing models is the ratio of their marginal likelihoods (the denominators of (2.11)). For many models, this value takes the form of an intractable integral; hence the exact calculation of the Bayes factor is impossible in these scenarios. There exists several methods estimating the Bayes factor, including the Harmonic Mean Estimator[86], Bridge sampling[87], Path sampling[88], the method of Chib (for both Gibbs Sampling[89] and Metropolis-Hastings[90]) and the Savage-Dickey Density Ratio (SDDR)[91].

Harmonic Mean Estimator

The method[86] which seems immediately obvious to estimate the marginal likelihood of a given model is to generate likelihood values at each MCMC step from the posterior samples, before taking the mean likelihood. This is the Harmonic Mean Estimator (HME).

However, it has been demonstrated[92][93] that the HME does not give accurate results, since the estimates it generates are dominated by the parts of posterior probability space where the likelihood is small, leading to infinite variance of the estimates. Additionally, the HME is insensitive to the prior distribution by nature of it being derived from sample of the posterior, which is by its very nature insensitive to the prior distribution. However it is well known[72] that the marginal likelihood should be

extremely sensitive to the prior. Hence the HME, despite its simplicity, does not yield accurate inference of marginal likelihood values.

Savage–Dickey Density Ratio

Another simple-to-apply method is the Savage–Dickey density ratio (SDDR).

Suppose we have the full model M_F with some set of parameters θ, with prior distribution $P(\theta)$. Nested within this model is a special case, the null model M_N, where one of the elements of θ, θ_0, is equal to some value, say zero.

Then we observe data y.

In order to calculate the model probability $P(M_F)$ we need the marginal likelihood ratio of the null model versus the full model:

$$\frac{P(y|M_N)}{P(y|M_F)} \tag{2.23}$$

The Marginal Likelihood of the full model is:

$$P(y|M_N) = \int P(y|\theta, M_N) \cdot P(\theta|M_N) d\theta \tag{2.24}$$

$$= \int P(y|\theta, \theta_0 = 0, M_F) \cdot P(\theta|\theta_0 = 0, M_F) d\theta \tag{2.25}$$

$$= P(y|M_F, \theta_0 = 0) \tag{2.26}$$

$$\overset{Bayes}{=} \frac{P(\theta_0 = 0|y, M_F) \cdot P(y|M_F)}{P(\theta_0 = 0|M_F)} \tag{2.27}$$

Then rearrange to get:

$$\frac{P(y|M_N)}{P(y|M_F)} = \frac{P(\theta_0 = 0|y, M_F)}{P(\theta_0 = 0|M_F)} \tag{2.28}$$

All that is needed for the estimation of Bayes factors with SDDR is the density of the prior ($P(\theta_0 = 0|M_F)$) and posterior ($P(\theta_0 = 0|y, M_F)$) in the full model at $\theta_0 = 0$.

Chib's Methods

The methods of Chib applied to the output from Gibbs samplers[89] and Metropolis–Hastings samplers[90] estimate the marginal likelihood by calculating the ratio of estimated posterior density against the densities of the prior and likelihood at a high

density point in the posterior, such as a modal value. Crucially, the accuracy of these methods depends only on the accuracy of the posterior density at the chosen high density point.

Bridge Sampling and Path Sampling

Bridge sampling[87] and its generalisation, path sampling[88] are iterative methods which approximate the Bayes factor by "bridging" with posterior density with a third bridge density. It should be noted that these methods require that each model being considered to be fitted separately. Furthermore accuracy of the estimate of the Bayes factor is dependent on there being many more (an order of magnitude more) posterior samples generated than would be generally required for accurate estimation of e.g. the posterior mean[94]. These two factors mean that estimation of the Bayes factor using bridge/path sampling may be much more computationally expensive versus SDDR.

In their tutorial[95] on bridge sampling, Gronau et al. make a number of recommendations for those wishing to derive approximations for the marginal likelihood. In particular they suggest the use of the SDDR in cases where one of the models being compared has the other (or others) nested within in it. Mootoovaloo et al.[96] make similar recommendations, going further to suggest the creation of "supermodels" in which all models of interest are nested so that SDDR can be applied.

2.3.10 Bayes Factors for Differential Expression

In the context of a generic MCMC sampler, the methods of most importance are the Harmonic Mean Estimator and the Savage-Dickey Density Ratio, since they can easily be applied to any technique which generates posterior samples. The others either require access to internal values in a sampling framework (as is the case for Chib's methods[89],[90]), or are not accurate in cases where the posterior densities of the two models are not nearby in the sample space (bridge and path sampling), which is likely to be the case for models for quantitative proteomics when proteins are differentially expressed.

The use of Bayes factors for differential expression has seen some limited application in genomics[97][98] but these have been limited to estimation of the marginal likelihoods via the harmonic mean estimator[86] which, as discussed above does not give accurate estimates of the marginal likelihood[92][93].

A related approach for model selection is Reversible-Jump MCMC (RJMCMC)[99] where one of the parameters in the model is a binary or categorical model indicator

variable. Effectively allowing for MCMC moves to be made between and within discrete models, RJMCMC has been utilised in genomics[68] and in transcriptomics[100] to perform differential expression analysis by generating a probability of differential expression.

RJMCMC cannot generally be tackled by HMC samplers, requiring the use of a discrete model indicator variable which is not compatible with the Hamiltonian dynamics. Also, in cases where one model is much favoured over others, accurate estimation of the Bayes factor through RJMCMC is difficult, since the small jumping probability leads to low sample sizes for the other model (or models)[101].

2.4 Conclusions

Many of the methods for protein quantification and differential expression analysis considered above are frequentist in nature The equivalent Bayesian models to these frequentist models would be ones where all parameters have uniform prior distributions. A key advantage over frequentist approaches for inference that Bayesian methods have is that they can incorporate detailed prior information where appropriate, be this from prior knowledge regarding the likely values of parameters or using hierarchical models (or approximations to hierarchical models through empirical Bayes methods) to propagate uncertainty about the probable values of parameters in a model. Furthermore, Bayesian methods such as MCMC sampling have been developed to allow for credible intervals to be generated for all parameters, which are directly interpretable as the range of likely values for those parameters. Frequentist approaches on the other hand tend to rely on asymptotic approximations such as (restricted) maximum likelihood estimation to fit statistical models.

Quantitative proteomics provides a number of challenges that Bayesian methods are well-suited to tackle. The amount of data available for any one protein can vary across an experiment, not just in the number of peptides and features that are detected and quantified, but in the numbers of missing data points. Hence, it is likely to be beneficial to leverage Bayesian hierarchical modelling to effectively borrow strength from proteins for which there is sufficient data in order to make more robust inference about proteins for which there is less data available. Furthermore, where the resulting protein quantification estimates are uncertain, this additional information should be used to improve differential expression testing and multiple hypothesis correction.

2.4.1 Aims of This Thesis

The following chapters of this thesis explore a number of identified challenges in quantitative proteomics: robust estimation of protein quantifications from peptides and feature level data; identifying differentially expressed proteins and estimation of the false discovery rate; and inclusion of shared peptides in the process of protein quantification. Broadly, this work aims to evaluate existing methods for protein quantification and differential expression analysis and and improve upon them by applying a Bayesian approach, namely by evaluating the precision and recall of differentially expressed proteins for these methods, additionally evaluating the methods' ability to estimate the false discovery rate. This is achieved through the analysis of spike-in datasets which, through the spiking-in of proteins from one species at differing concentrations into a constant background of proteins from another species, provide suitable surrogate problems to clinical datasets in which the ground truth can be determined.

Chapter 3

The BayesProt Model

3.1 Introduction

The BayesProt[102][2] pipeline for protein quantification is a package for the R programming language[103] which generates protein quantification estimates from mass spectrometry data using Bayesian inference to fit a mixed-effects model to feature-level intensities, concurrently estimating the reliability of individual peptides and individual features.

Input feature intensities can come from one of a number of popular pre-processing tools: Sciex's ProteinPilot software; MaxQuant[28]; Thermo's ProteomeDiscoverer; MSStats[32]; and Waters' Progenesis Software.

The mixed-effects model incorporates random effects to account for variability at multiple levels: variability of peptides across samples (due to poor digestion for example); and variability of features across assays (for example, due to contaminants). Experimental design is specified with a simple data table in R, where assays are assigned to specific samples. The distinction between assays and samples is made so that pure technical replicates of the same digested sample can be used to better infer the proportion of observed variance due to digestion and labelling versus the proportion due to technical variation of the mass spectrometry and chromatography process.

Inference is achieved with Markov chain Monte Carlo (MCMC) sampling performed using the MCMCglmm R package[104]. The model is applied in two stages: firstly, the model is fitted to proteins with at least three peptides and at least three features. From the results of these proteins an empirical Bayes procedure is then used to calculate informative priors for peptide and feature variances. These informative prior distributions are then used in the second stage model. The second stage model is then fitted to all

other proteins (those with two or fewer peptides or features). The informative prior distributions allow the model to borrow strength from the proteins with more available data.

Normalisation is then performed, using the protein-level quantification estimates to infer the median fold-change between each assay and either a single reference assay or the mean of multiple assays. The median fold-changes across all proteins are then used to normalise each assay.

The posterior median protein quantification estimates for each protein are output for each assay with an associated measure of uncertainty. These protein quantification estimates can be output for further processing by other tools. Optionally, differential expression analysis can be performed by BayesProt by wrapping external methods which take into account the uncertainty of protein quantifications.

Earlier versions of the software have been applied for proteomic analysis in a number of studies[2][1][105]. This chapter concerns the current version of the model and software.

Section 3.2 describes the statistical model used in BayesProt, explaining the intended function of each of the parameters. It then goes on to explain how an empirical Bayes procedure is used to inform the prior distributions on proteins for which there is little available data.

Section 3.3 describes the methods used to validate the model, namely the spike-in data sets used, the seven progressively more complex versions of the base model (including an extension of the empirical Bayes method) and the different techniques for differential expression and FDR estimation.

Section 3.4 shows results demonstrating the recall abilities of each of the combinations of the above methods for differential expression and FDR estimation for the six versions of the base model.

In Section 3.5, results are presented showing the differences in FDR control and subsequent "calibrated recall", a measure of real-world performance.

Finally, Section 3.6 summarises with concluding remarks and discusses potential improvements, some of which are motivations for later chapters.

3.2 BayesProt Model

The BayesProt model is fitted to each protein in the data set separately in two stages. The subsetting of proteins is based on two factors: the number of identified peptides and the number of identified features. By default, thresholds of two peptides and two

features determine whether a protein is analysed in the first or second stage. Proteins with more peptides than the peptide threshold and more features than the feature threshold are analysed in the first stage. Conversely, proteins with fewer peptides than the peptide threshold or fewer features than the feature threshold are analysed in the second stage. Alternative thresholds can be set by the user for the number of peptides and features required for a protein to be analysed in the first or second modelling stage.

3.2.1 First Stage Model

The analysis begins by fitting the base model to all proteins that have been assigned to the first stage. The model is as follows:

$$\lambda_{j,k,s,f} = \exp(\beta_f^{\text{Feature}} + \beta_k^{\text{Assay}} \epsilon_{s,j}^{\text{Sample}} + \epsilon_{k,f}^{\text{Residual}}) \tag{3.1}$$

$$y_{j,k,s,f} \sim \text{Poisson}(\lambda_{j,k,s,f}) \quad \text{for } y_{j,k,s,f} > 0 \tag{3.2}$$

$$P(y_{j,k,s,f} < C_f) = \sum_{x=0}^{C_f - 1} [\text{Poisson}(x|\lambda_{j,k,s,f})] \quad \text{for } y_{j,k,s,f} = 0 \tag{3.3}$$

where $y_{j,k,s,f}$ is the observed ion count (or XIC intensity) of a feature f belonging to a peptide j in assay k, sample s and C_f is the minimum non-zero count/intensity for a feature f. This is known as a generalised linear mixed-effects model with a Poisson likelihood and a log link function, with fixed effects β^{Feature} and β^{Assay} for feature and assay. The fixed effects are given normal prior distributions:

$$\beta_f^{\text{Feature}} \sim \text{Normal}(0, 1 \times 10^{10}) \tag{3.4}$$

$$\beta_k^{\text{Assay}} \sim \text{Normal}(0, 1 \times 10^{10}) \tag{3.5}$$

Also included are a peptide deviation random effect term $\epsilon_{s,j}^{\text{Sample}}$ for sample s and peptide j whose variances are shared across peptides:

$$\epsilon_{s,j}^{\text{Sample}} \sim \text{Normal}(0, \sigma_j^{\text{Peptide}}) \tag{3.6}$$

$$\tag{3.7}$$

This parameter captures the deviations of peptide j from the consensus protein-level quantifications. For the proteins analysed in the first modelling stage the random effects

are given parameter-expanded priors to improve mixing of the model: where variance parameters are sampled near zero the Gibbs sampler is forced to make smaller steps for the latent random effects, resulting in slow mixing. Decoupling the variances from the random effects by adding additional, unidentified parameters to the model allows the sampler to explore the parameter space more efficiently[104]. In the first modelling stage, the per-peptide standard deviations across peptide deviations are assigned half-Cauchy priors (as recommended by [106]):

$$\sigma_j^{\text{Peptide}} \sim \text{Half-Cauchy}(\mu = 0, \tau = 25) \tag{3.8}$$

$$\tag{3.9}$$

with location parameter μ and scale parameter τ.

Finally, a so-called residual error term is included (in reality, another random effect), $\epsilon_{k,f}^{\text{Assay,Feature}}$ for assay k and feature f whose variances are shared across features:

$$\epsilon_{k,f}^{\text{Residual}} \sim \text{Normal}(0, \sigma_f^{\text{Feature}}) \tag{3.10}$$

which, like the peptide-sample random effect, captures feature-level deviations from the consensus protein-level quantifications.

Finally, the per-feature residual variance is assigned a scaled-inverse-χ^2 prior distribution:

$$\left(\sigma_f^{\text{Feature}}\right)^2 \sim \text{Scale-Inv} - \chi^2(\nu = 0.02, \tau^2 = 1) \tag{3.11}$$

The model is summarised in the Bayesian network/plate diagram in Figure 3.1.

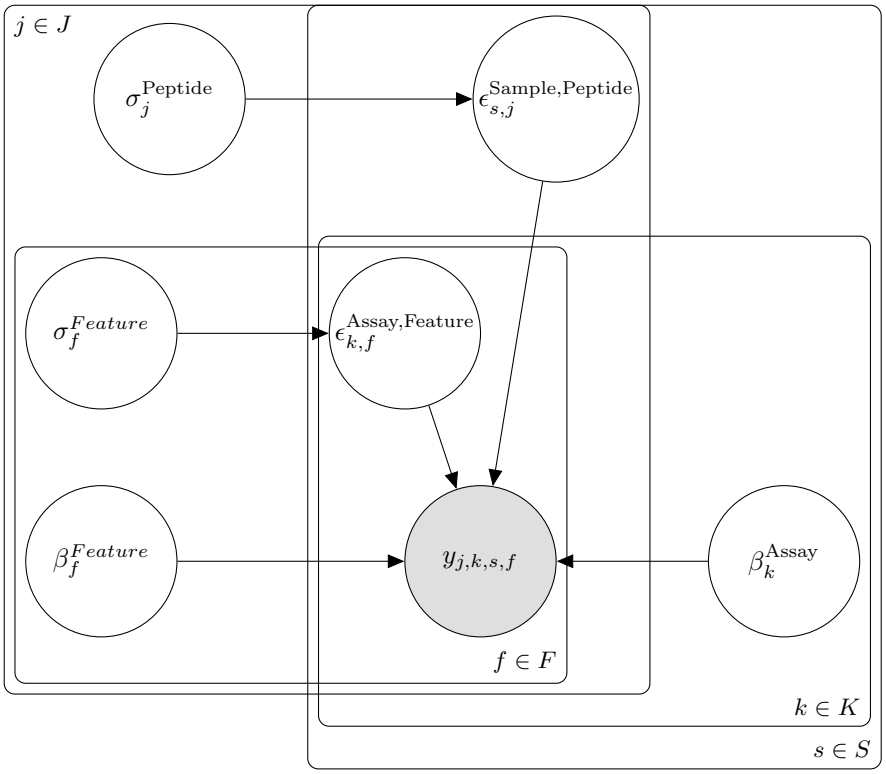

Figure 3.1: Bayesian plate diagram representation of the BayesProt model for a single protein. F denotes the set of features; J, the set of peptides; K, the set of assays; and S the set of samples. Hyperparameters are not included for clarity.

Explanation of Parameters

The feature fixed effect β_f^{Feature} allows for the model to account for systematic difference in the relative intensity of each LC-MS feature (or iTRAQ/TMT spectrum as applicable).

The per-sample random effect $\epsilon_{s,j}^{\text{Sample}}$ with per-peptide variance $\sigma_j^{\text{Peptide}}$ allows the model to down-weight peptides whose quantification pattern across the assays does not agree with the consensus. Hence, peptides that are highly variable (say, as a result of the peptides' poor digestibility), have a smaller contribution to the overall protein quantification estimates.

Similarly, the per-feature residual variance $\sigma_f^{\text{Feature}}$ allows the model to down-weight highly variable features whose quantification pattern does not agree with the consensus.

The quantity of interest for further downstream analysis is the estimated log-fold-change between each assay in the experiment captured in the distribution of the corresponding fixed effect β_k^{Assay}.

The model is fitted using the Gibbs MCMC sampler provided by the MCMCglmm[104] R package, which draws samples from the joint posterior distribution. Multiple MCMC chains are fitted for each protein — by default four. Posterior samples for each of the above parameters are saved for each protein. Analysis then moves to an intermediate empirical Bayes step.

3.2.2 Informative Priors on Peptide and Feature Variances Through Empirical Bayes

For proteins with few peptides, it is difficult to ascertain whether the observable changes in the intensity of peptides are due to random variability due to (poor) digestion or due to true differences in the abundance of the underlying proteins.

BayesProt borrows strength across the population of proteins, using the data from proteins with a sufficient number of peptides to generate an informative prior for peptide variance which is then subsequently used for the analysis of proteins with fewer peptides.

This borrowing of strength is accomplished by employing a reduced empirical Bayes approach to inform the prior distributions on peptide and feature variances for those proteins with fewer peptides or features.

Initially the model fits a variance for each peptide for those proteins for which there is sufficient data. The BayesProt model is fitted to these proteins in the first modelling stage and the technical variation of each peptide across digested samples is determined.

A scaled-inverse-χ^2 distribution is then fitted to the inferred median variance of peptides across the data set, using the fitdistrplus[107] R package. This scaled-inverse-χ^2 distribution is subsequently used as an informative prior on peptide technical variance for the analysis of proteins with fewer than three peptides. These proteins, for which there is not sufficient data to estimate the technical variance, will then have their posterior peptide variance being dictated almost entirely by the prior.

Alongside informing the peptide variance, we can also perform the same procedure with the residual feature variances for those proteins with very few observed features: by default, like peptide variances, informative priors are only applied to proteins with fewer than three features. Again, the result is that protein quantification estimates are robust to low-quality features, even when a protein has very few features from which to make inferences.

An example of the fitted scaled-inverse-χ^2 distributions used as informative priors is shown in Figure 3.2.

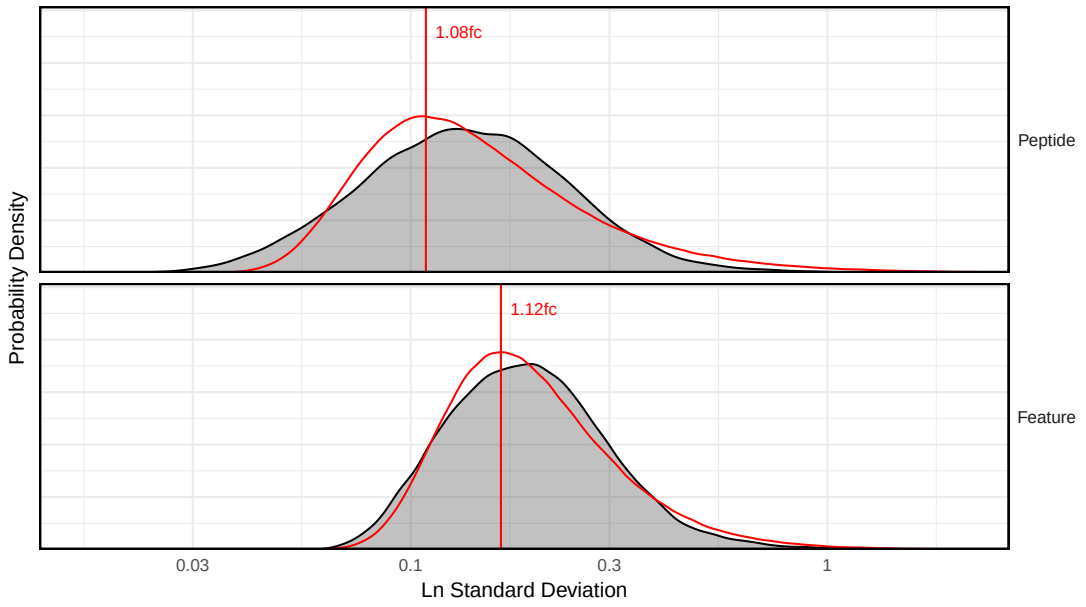

Figure 3.2: Example of fitted distributions for informative priors on peptide variance (top) and feature variance (bottom). Kernel density estimates fitted to the median variances across all peptides and features are shown in grey. The fitted scaled-inverse-χ^2 distributions are shown in red, with the median of the sampled median variances shown as red vertical lines. This plot is one of a number outputted by BayesProt.

3.2.3 Second Modelling Stage

The analysis then proceeds to the second modelling stage. Here, the same model as in the first stage is fitted to the proteins with fewer peptides or features, this time using the informative scaled-inverse-χ^2 priors for peptide variance and feature variance that were estimated by the empirical Bayes.

3.2.4 Normalisation

As described in Section 2.1.8, normalisation is required to correct for any systematic biases between samples that are not part of the experimental design, for example, due to sample handling. Median normalisation attempts to correct for these biases by basing normalisation on only those proteins that are unchanging between sample groups by scaling intensities to equalise the medians. After the second modelling stage of BayesProt is complete, a Bayesian median normalisation procedure is performed using the inferred per-assay protein quantifications for each protein. Firstly, the median value for each assay/protein combination is calculated for each MCMC sample. Then the distribution of medians across proteins for each assay is used as a robust estimator for the normalisation effects: normalised protein quantifications are calculated based on the distribution of normalisation effects for each assay, rather than a single value, by using the inferred normalisation at each MCMC iteration. An example of these distributions of medians is illustrated in Figure 3.3.

The resulting protein quantification estimates that are output consist of the MCMC samples for each of the normalised assay-level quantifications for each protein.

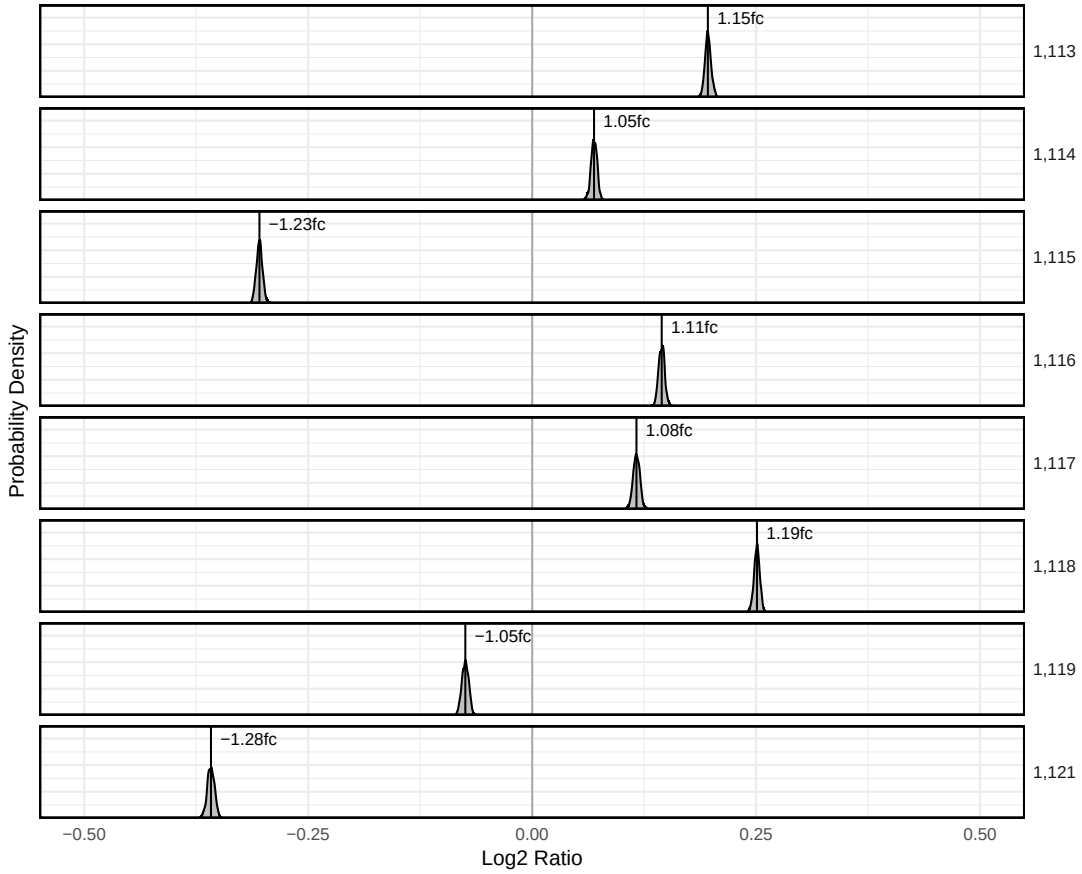

Figure 3.3: Example of inferred normalisation effects output by BayesProt for the spike-in data used in Section 3.3. The density plot for each assay is shown as a separate row. Kernel density estimates of the medians of each protein are shown in grey for each assay. The means for each assay are shown as black vertical lines with corresponding inferred fold-change.

3.2.5 Differential Expression Analysis

Currently, the BayesProt package provides the ability to perform differential expression analysis with a number of external methods out-of-the-box, which properly account for the uncertainty in the protein quantifications estimates. These quantification estimates can be summarised by taking the posterior median for each protein in each assay across all the MCMC samples as a robust estimator of the mean. The uncertainty in these quantifications can be summarised with the calculation of the median absolute deviation, a robust estimator for the posterior standard deviation, across the MCMC samples. These protein quantifications are suitable for subsequent differential expression analysis by BayesProt's built-in methods or with other downstream tools for statistical testing or, for example, biological pathway analysis.

Assays can be assigned to different conditions (treatment groups) which are used as comparison groups for differential expression. Using the design table specified by the user, the protein quantifications for each protein and assay are assigned to the treatment groups so that significant differences in protein quantification across the treatment group can be determined by one of a number of different methods.

Meta-analytic T-Test

The metafor[108] R package performs a random-effects meta-analytic t-test, which differs from a standard Student's t-test by allowing users to supply a measurement error for each observation. BayesProt approximates the standard error by the posterior standard deviation of the protein quantifications, and supplies these to metafor to perform statistical testing. Metafor fits a mixed-effects model, by default using Restricted Maximum Likelihood (ReML), estimating the underlying residual variance so that the true effects can be estimated. A t-statistic is calculated to generate a p-value for a "Condition" covariate.

Bayesian Test

Another included differential testing method utilises the MCMCglmm[104] R package, fitting a simple mixed-effects model through MCMC. By default, this model is configured as a Bayesian analogue to a Student's t-test, but with the additional option for the protein quantification uncertainty to be taken into account:

$$\mathbf{y} \sim \text{Normal}(\mathbf{X} \cdot \boldsymbol{\beta}, \sqrt{\sigma^2 + \mathbf{SD}^2}) \tag{3.12}$$

where $\mathbf{X}$ denotes a design matrix, $\boldsymbol{\beta}$ the vector of estimated "Condition" covariates and $\mathbf{SD}$ the posterior standard deviations of the protein quantifications.

False Discovery Rate Estimation

The p-values generated by the metafor package are suitable for multiple testing correction by either Benjamini–Hochberg[60] (available as the `p.adjust` command in R) or Storey's method[65] (implemented in the qvalue[109] R package).

These frequentist methods for FDR estimation are not applicable to the differential expression estimates generated by the Bayesian t-test. The ash[70][110] R package (described in further detail in Section 2.2.3) can be used to generate FDR estimates for each protein, using the estimated effects for differential expression, plus standard errors and degrees of freedom, calculated by either the meta-analytic t-test or the Bayesian t-test. For the Bayesian t-test, these are approximated by fitting t-distributions to the posterior samples of the condition effects.

Both the meta-analytic t-test and Bayesian t-test have the added benefit (over e.g. a simple Student's or Welch's t-test) of utilising the uncertainty of the protein quantifications generated by the model. The ash R package, which leverages this additional information, has been shown to demonstrate an increase in power over older methods such as BH and Storey's q-value[61].

Summary

In summary, BayesProt fits a Bayesian model to observed feature-level intensities (e.g. ion counts or XIC areas in the case of a label-free analysis), taking the variability of individual peptides and features into account in order to down-weight the contribution to the resulting protein quantification estimates of more variable peptides and features. Additionally, it provides a number of options for performing differential expression analysis to determine which proteins in the input data are likely to be changed between the treatment groups specified.

This chapter aims to evaluate the BayesProt model's ability to provide more accurate protein quantification estimates, validating both the model itself and the included differential expression testing methods through the analysis of spike-in data sets.

3.3 Methods

The BayesProt model for protein inference was validated by demonstrating increased precision and recall of differentially expressed proteins in a spike-in data set, comparing the results garnered by applying t-tests and meta-analysis t-tests to BayesProt's protein quantifications, comparing to the results from applying those same methods for differential expression testing to protein quantifications yielded by other existing methods.

3.3.1 Spike-in Dataset

The spike-in data set used for validation was provided by Dr Richard Unwin of the University of Manchester. The data set consists of a 4 vs 4 iTRAQ experiment in a single 8-plex with two sample groups, A and B. All eight samples contained 75µg of protein from a rat kidney lysate. The rat kidney lysate was taken from an earlier proteomics study, for which ethical approval was given by the relevant NHS trusts and the University of Manchester. Samples were spiked with *Escherichia coli* (E. coli) proteins from a whole cell lysate. Samples in group A were spiked with 10µg of E. coli protein, and samples in group B were spiked with 16µg of E. coli protein. In addition to the E. coli, a number of individual proteins were spiked-in at different concentrations, detailed in Table 3.1. Each sample was made up independently. Digestion and processing was then performed according to the lab's standard protocols (detailed in [1]). Crucially, this differs to many other spike-in data sets, where samples within a group are technical replicates. This spike-in data set mimics biological replicates in clinical studies where samples are necessarily prepared and digested separately. Mass spectrometry analysis was then performed using a 6600 TripleTOF instrument. Peak extraction, peptide identification, protein grouping and iTRAQ reporter quantification was performed using ProteinPilot v5.0.

Different normalisation procedures can have a pronounced effect on the number of differentially expressed proteins identified as possible discoveries. This is especially true for experiments such as the spike-in experiment considered here, where the distribution of true fold-changes is not symmetric; the median fold-change in this data set is non-zero. Since the effect of different normalisation methods is not being evaluated here, median normalisation was performed using only the rat background proteins to ensure that any observed differential expression is not simply an artefact of incorrect normalisation.

Table 3.1: Detail of proteins in the spike-in validation data set.

Protein	Fold-Change	log 2 Fold-Change
RAT	1.00	0.0
ECOLI	1.60	0.678
sp|P00004|CYC_HORSE	0.800	−0.322
sp|P68082|MYG_HORSE	2.00	1.0
sp|P02754|LACB_BOVIN	0.800	−0.322
sp|P02666|CASB_BOVIN	1.333	0.415
sp|P00330|ADH1_YEAST	0.333	−1.58
sp|P01012|OVAL_CHICK	1.333	0.415
sp|P06278|AMY_BACLI	2.00	1.0
sp|P00432|CATA_BOVIN	1.870	0.900
sp|P02760|ALBU_BOVIN	0.667	−0.585
sp|Q29443|TRFE_BOVIN	2.00	1.0
sp|P46406|G3P_RABIT	0.750	−0.415
sp|P00489|PYGM_RABIT	0.750	−0.415
sp|P00698|LYSC_CHICK	0.667	−0.585
sp|P00711|LALBA_BOVIN	0.667	−0.585
sp|P00915|CAH1_HUMAN	0.667	−0.585
sp|P08603|CFAH_HUMAN	0.600	−0.737
sp|P08603-2 |CFAH_HUMAN	0.5	−1.0

3.3.2 BayesProt Configuration

In order to validate the BayesProt model, a number of versions of the model were evaluated, differing in terms of the choice of variance parameter for the peptide random effect and the residual variance:

- Peptide modelling — Independent variances for each peptide; single variance per protein; no peptide random effect
- Feature modelling — Independent residual variances for each feature; single residual variance per protein.

Each of the six combinations of these two factors were evaluated.

One natural extension to the BayesProt model is to apply the empirical Bayes (EB) priors on peptide and feature variance to *all* proteins. A similar approach was successfully applied in [31], though only applied to residual variances and using the EB procedure provided by the limma[57] R package, in order to stabilise the uncertainty of protein quantifications. The BayesProt model was also run with the default configuration, with the added modification that the second stage model was extended to include all proteins; therefore the informative priors on peptides and feature variances calculated in stage one of the model are subsequently applied to all proteins.

The configurations used in the validation are as follows:

- No peptide random effect; single residual variance
- No peptide random effect; per-feature residual variance
- Single peptide variance; single residual variance
- Single peptide variance; per-feature residual variance
- Per-peptide variance; single residual variance
- Per-peptide variance; per-feature residual variance — default BayesProt configuration
- Per-peptide variance; per-feature residual variance — full empirical Bayes

For each of these configurations, BayesProt was run five times, initialised with different random seeds.

3.3.3 Methods for Protein Quantification

Eight different methods for protein quantification were tested.

ProteinPilot + Student t-test

The baseline method with which we here make comparisons is the protein quantification estimates calculated by the ProteinPilot software. These protein-level quantification ratios are calculated as simple averages of the ratios of all peptides belonging to each protein. Notably, this method does not factor in the uncertainty of the individual peptide ratios.

MSstatsTMT + Moderated t-test

A more advanced method available for the analysis of iTRAQ data is the MSstatsTMT[111] R package, an extension of the MSstats[32] software for isobaric labelled data. The ProteinPilot PeptideSummary data was adapted to allow MSstatsTMT to be used; MSstatsTMT currently cannot import data from ProteinPilot. A linear model is fitted using ReML to summarise the feature-level quantifications up to protein-level quantifications. A moderated t-statistic is then calculated using limma's[44] empirical Bayes procedure to give a p-value for each protein.

BayesProt Protein Quants + Student t-tests

Similarly, simple protein quantification values are calculated from the MCMC samples generated by BayesProt by taking the posterior median for each assay. Student's t-tests are then performed on these median quantification estimates to calculate a p-value for each protein.

BayesProt + Meta-analytic t-tests

The metafor R package is also used to perform differential expression testing, as described in Section 3.2.5.

BayesProt + MCMCglmm (Equal Variances)

Further improvement over the meta-analytic t-tests is gained through the use of a Bayesian test using the MCMCglmm R package to perform further MCMC on the

generated proteins quantifications, again taking the uncertainty of each quantification into account.

For protein quantifications y_i with associated posterior standard deviations SD_i, the following is sampled from using MCMCglmm:

$$\boldsymbol{\mu} \sim \text{Normal}([0,0], \mathbf{I} \cdot 10^2) \tag{3.13}$$

$$\sigma^2 \sim \text{Scale-Inverse-Chi-Squared}(0.02, 1) \tag{3.14}$$

$$y_i \sim \text{Normal}(\mathbf{X}\boldsymbol{\mu}, \sigma + SD_i) \tag{3.15}$$

MCMC samples corresponding to the condition effect for each protein are then summarised by the medians and median-absolute-deviations.

BayesProt + MCMCglmm (Unequal Variances)

MCMCglmm is also used to fit an unequal variances model akin to Welch's t-test:

$$\boldsymbol{\mu} \sim \text{Normal}([0,0], \mathbf{I} \cdot 10^2) \tag{3.16}$$

$$\sigma_c^2 \sim \text{Scale-Inverse-Chi-Squared}(0.02, 1) \tag{3.17}$$

$$y_i \sim \text{Normal}(\mathbf{X}\mu, \sigma_c + SD_i) \tag{3.18}$$

where σ_c is a per-condition variance, rather than a single variance across all conditions as used above. The unequal variances t-test has been found to be more reliable than Student's equal variances t-test[112]. Furthermore, differential expression is likely to affect sample composition due to, for example, ionisation competition between proteins affecting all observed intensities. Hence, it is likely that variances are naturally unequal between populations. The effect sizes for each protein are again summarised by the median and median-absolute-deviation.

Both the Bayesian t-tests with equal and unequal variances were also evaluated without the inclusion of the estimated protein quantification uncertainties.

3.3.4 False Discovery Rate Estimation

Six methods for false discovery rate (FDR) estimation were evaluated. For methods producing p-values, two main, popular methods exist for multiple hypothesis correction: the Benjamini–Hochberg procedure[60] and Storey's q-value[65].

Benjamini–Hochberg

For the Student's t-test, Welch's t-test and Metafor t-test, Benjamini–Hochberg correction is applied to the generated p-values to calculate adjusted p-values using the `p.adjust` function in R.

Storey's q-value Method

Similarly, Storey's q-value method is applied to the same p-values; crucially this does not change the ordering of the list of proteins, only the estimate of FDR.

For the purposes of this validation the proportion of null p-values, π_0, is not specified since in most real applications this would not be a known value. Instead π_0 is estimated by the qvalue[109] package by fitting a uniform distribution to the distribution of p-values far from zero, based on the rationale that p-values of non-differentially expressed proteins would be distributed uniformly[113].

The ash R package

Finally, the ash R package is applied to all the above differential expression methods. It allows for a number of options to be set. Of particular interest are the `df` and `mixcompdist` parameters. Parameter `df` determines the degrees of freedom used for calculating t-statistics. This value can be set to ∞ which forces ash to use a Z-statistic rather than a general t-statistic. The `mixcompdist` parameter determines the type of distribution used as mixture components for ash's estimation of the population-level distribution of fold-changes. Here, the uniform and half-uniform distributions were selected for evaluation, since these correspond to symmetric and non-symmetric distributions of fold-changes and provide the most flexibility for setting the *df* parameter; choosing a normal or half-normal distribution for the mixture components constrains the choice of *df* to infinity.

Four configurations of ash were evaluated: using a uniform component distribution and *df* = infinity; using a uniform component distribution and *df* specified for each protein; using a half-uniform component distribution and *df* = infinity; using a half-uniform component distribution and *df* specified for each protein. The use of uniform and half-uniform distributions for allows for the degrees of freedom of the estimates to be set to a value other than infinity (which implies normality); [61] notes that ash is sensitive to this quantity.

3.3.5 False Discovery Proportion and True Discoveries

For each combination of the above methods, a list of proteins ordered by estimated FDR is output. Since the fold-change of the background rat proteins is known, for each list false discovery proportions (FDPs) can be calculated at each point along the ordered list, as the cumulative mean of the number of false positives:

$$\text{FDP}_i = \frac{1}{i}\sum_{j=1}^{i} 1 - \text{DE}_j \tag{3.19}$$

where DE_i is a binary indicator of whether the protein i is a true positive (a non-rat protein). This is then interpolated by taking the cumulative minimum of the FDP further down the list:

$$\text{FDP}_i = \min(\text{FDP}_{j \geq i}) \tag{3.20}$$

The false discovery proportion is equivalent to $1 - \text{Precision}$.

The number of true discoveries at each point is similarly calculated as the cumulative number of true positives:

$$\text{True Discoveries}_i = \sum_{j=1}^{i} \text{DE}_j \tag{3.21}$$

3.3.6 Mixing

For each of the BayesProt runs used for validation, for each protein 1024 MCMC samples were drawn from four MCMC chains (256 from each) with 256 warm-up iterations per chain using the MCMCglmm R package. In order to justify the number of MCMC samples and warmup iterations used for inference, the Gelman–Rubin $\hat{R}$ diagnostic was used. Prior to validation the spike-in data set was used to verify that this number of MCMC samples and warmup iterations was sufficient to ensure the convergence of the MCMC chains. Firstly, the number of warm-up (burn-in) iterations was varied. By default, BayesProt generates 1024 MCMC samples for each protein across four MCMC chains with 256 warm-up iterations. BayesProt was also run with 128 and 64 warm-up iterations per chain. The total number of samples drawn was also varied. Using 64 warm-up iterations, the number of samples drawn from four chains was set to 1024 (the

default), 512, 256 and 128.

For each of these configurations, BayesProt was run five times with differing random seeds. The $\hat{R}$ statistic was calculated for each assay-effect posterior for each protein, the maximum $\hat{R}$ statistic across all assays was then recorded for each protein as a measure of convergence. Table 3.2 summarises the proportion of proteins with a maximum $\hat{R}$ statistic less than 1.2, a common threshold for convergence. Figures 3.4 and 3.5 present histograms of the maximum $\hat{R}$ for each protein for varying numbers of warmup iterations and total number of samples respectively.

Table 3.2: Summary of proportion of proteins with converged MCMC chains

		Proportion of Proteins with $\hat{R} < 1.2$	
Num. warm-up	Num. samples	1–2 peptides	3+ peptides
256	1024	0.951	0.994
128	1024	0.954	0.993
64	1024	0.953	0.993
64	512	0.930	0.971
64	256	0.883	0.888
64	128	0.730	0.578

The number of warm-up iterations used for MCMCglmm has little bearing on the convergence of the MCMC chains; since MCMCglmm uses a maximum likelihood estimation method to initialise the MCMC chains near to the typical set, only a small number of warm-up iterations are required, and so increasing this number has little effect on the convergence of the MCMC chains (see Figure 3.4).

However, the same cannot be said for the total number of samples. Reducing the total number of MCMC samples used for inference has a detrimental effect on the convergence of the MCMC chains (see Figure 3.5).

With 256 warm-up iterations and 1024 total samples, over 99% of proteins with three or more peptides and 95% of proteins with one or two peptides had $\hat{R}$ statistics of less than 1.2. Reducing this to 64 warm-up iterations and 512 samples had little effect on the convergence diagnostics, suggesting that any further increase over 1024 samples is likely to have diminishing returns (see Table 3.2). Thus, in the interest of minimising computation time this was deemed sufficient for the validation of the BayesProt model.

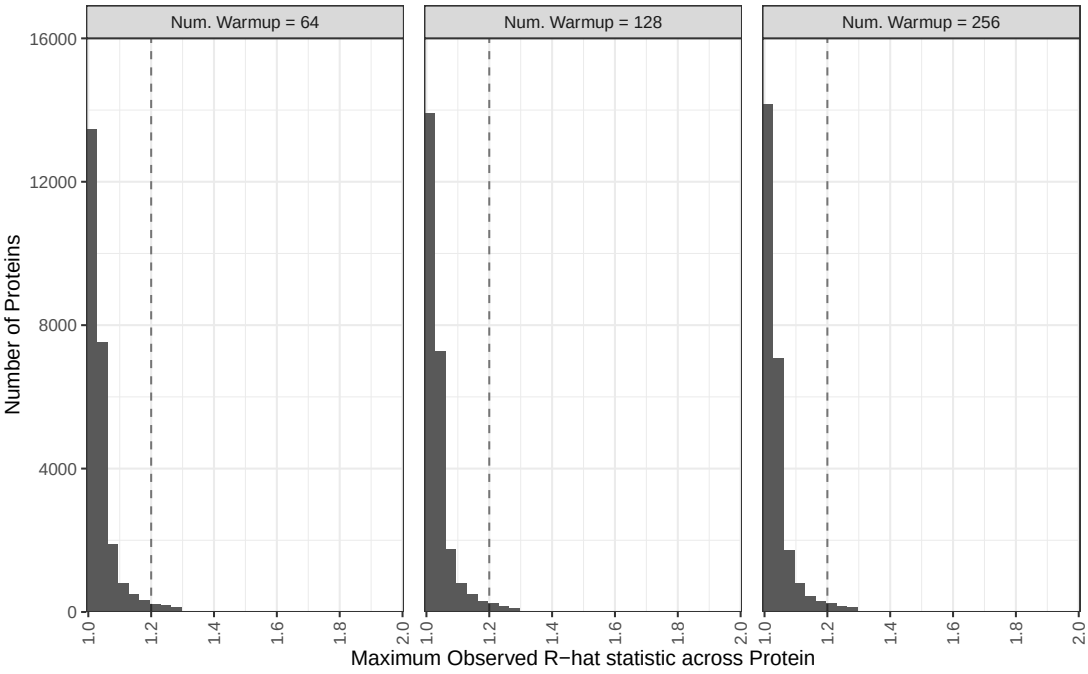

Figure 3.4: Histograms of maximum $\hat{R}$ statistic of each protein in the spike-in data used for validation for varying numbers of warm-up iterations across five runs of BayesProt for each configuration with differing random seeds for each run. The threshold $\hat{R} < 1.2$ is shown as a dashed line.

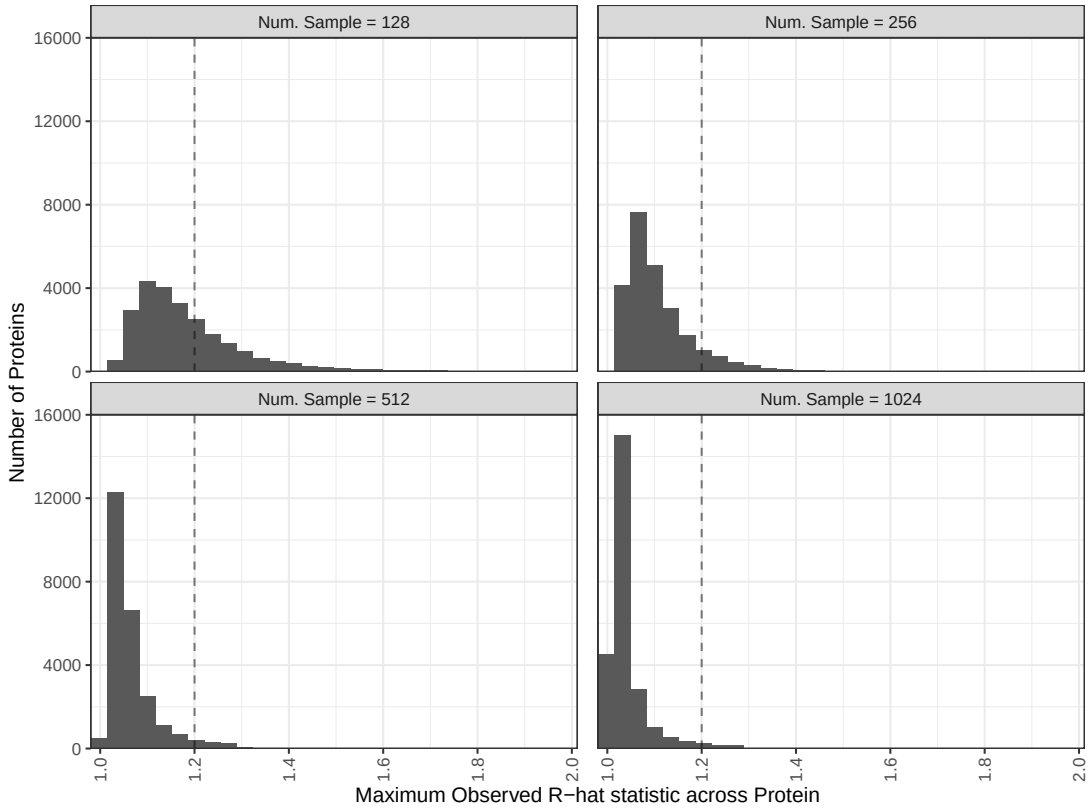

Figure 3.5: Histograms of maximum $\hat{R}$ statistic of each protein in the spike-in data used for validation for varying numbers of total samples across five runs of BayesProt for each configuration, with differing random seeds for each run. The threshold $\hat{R} < 1.2$ is shown as a dashed line.

3.4 Precision–Recall Curves

Firstly, we make comparisons in the ability of the tested procedures' abilities to correctly order proteins in the data set. Receiver operating characteristic (ROC) curves could be used for this task. However, ROC curves are not generally suitable for the analysis of inbalanced data[114] such as those generated by genomic or proteomic studies where the proportion of differentially expressed genes or proteins is much less than 50%; in the spike-in dataset considered here, there are 1164 spiked-in proteins versus 3861 background proteins. Instead more suitable precision-recall (PR) curves are presented for each method, showing FDP vs True Discoveries, as defined in Section 3.3.5; Since FDP is equivalent to $1 - \text{Precision}$, the y-axis is flipped so that $\text{FDP} = 0$ is at the top of the plot. The true discoveries metric is equivalent to recall except that recall is the *ratio* of true discoveries versus the total number of proteins.

Additionally, for the BayesProt-derived protein quantifications, we show results from increasingly complex versions of the model, going from a simple model with no modelling of peptides and a single residual variance per protein, up to the default configuration with a per sample random effect with per peptide variance and per feature residual variance, before demonstrating the effect of a full empirical Bayes approach.

In each of the plots below, the PR curves for ProteinPilot and MSstatsTMT are unchanging and hence act as points of reference, representing readily available and straightforward differential expression testing strategies that are commonly used in proteomics studies.

Also shown below are tables detailing the mean recall achieved by each method at each of three FDP thresholds: 1%, 5% and 10%. The mean recalls are calculated across the five runs of BayesProt and shown with calculated standard deviations.

It should be noted that in the plots and tables below, the choice of Storey's method for FDR estimation over Benjamini–Hochberg has no effect on the ordering of proteins; hence the curves and recall values are the same when these two methods are applied to any given quantification method. Meanwhile, since ash uses a population-level model of the distribution of fold-changes to calculate FDRs, the ordering of proteins will change between different configurations.

3.4.1 No Peptide Random Effect, Single Residual Variance

The simplest version of the model is that with no modelling of individual peptides' deviations from the consensus protein quantification pattern and a single residual variance across all features for each protein.

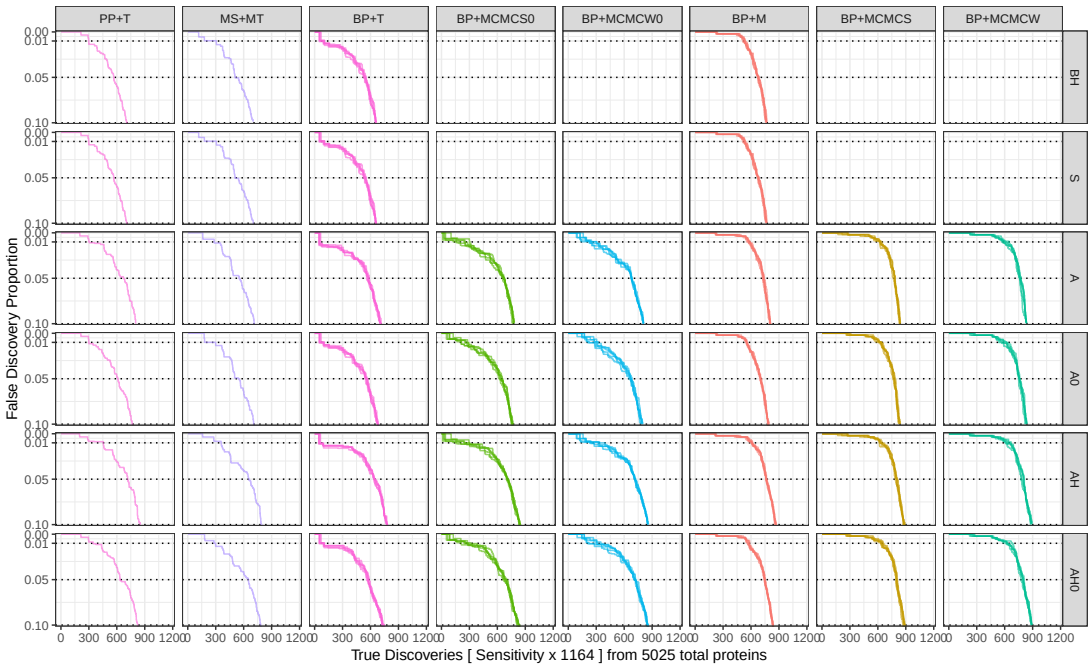

Figure 3.6: Precision-recall curves for single fraction spike-in data with no peptide random effect and a single residual variance for BayesProt. Results of repeat runs of BayesProt with differing random seeds are shown as separate lines.

Table 3.3: Mean recalls of tested methods for differential expression at multiple precisions with no peptide random effect and a single residual variance for BayesProt.

	Recall		
Method	1% FDP	5% FDP	10% FDP
PP+T+BH	299	567	707
PP+T+S	299	567	707
PP+T+A	306	671	802
PP+T+A0	300	612	765
PP+T+AH	456	716	851
PP+T+AH0	346	635	823
MS+MT+BH	307	507	702
MS+MT+S	307	507	702
MS+MT+A	283	553	713
MS+MT+A0	299	552	707
MS+MT+AH	358	656	794
MS+MT+AH0	299	651	783
BP+T+BH	84.6 ± 28.2	540.4 ± 11.1	657.8 ± 4.02
BP+T+S	84.6 ± 28.2	540.4 ± 11.1	657.8 ± 4.02
BP+T+A	53.2 ± 8.04	574.4 ± 10.7	707.8 ± 5.54
BP+T+A0	82.4 ± 26.2	555.6 ± 9.91	678.2 ± 2.17
BP+T+AH	51.2 ± 8.35	642.4 ± 5.73	778 ± 4.47
BP+T+AH0	83.6 ± 27.3	573.8 ± 9.5	736.2 ± 11.1
BP+MCMCS0+A	232.8 ± 55.5	667.4 ± 8.59	774.4 ± 8.53
BP+MCMCS0+A0	243.4 ± 29.6	639 ± 11.9	759.6 ± 7.5
BP+MCMCS0+AH	243.6 ± 62.5	707 ± 7.25	841.2 ± 6.06
BP+MCMCS0+AH0	269.6 ± 41.5	686.8 ± 16.2	825.6 ± 11.8
BP+MCMCW0+A	264.4 ± 38.4	679.8 ± 5.26	809.8 ± 3.63
BP+MCMCW0+A0	275 ± 52.1	668 ± 12.9	793.4 ± 7.83
BP+MCMCW0+AH	291.2 ± 49.6	715.8 ± 4.27	854.8 ± 4.44
BP+MCMCW0+AH0	280 ± 60.6	716 ± 9.27	852.4 ± 4.77
BP+M+BH	528 ± 10.5	679.2 ± 6.83	766 ± 3.67
BP+M+S	528 ± 10.5	679.2 ± 6.83	766 ± 3.67
BP+M+A	572.6 ± 11.4	733.8 ± 6.02	804.8 ± 5.26
BP+M+A0	552.2 ± 4.44	703.2 ± 5.54	784.4 ± 6.66
BP+M+AH	612 ± 4.64	763.2 ± 4.09	863.6 ± 7.89
BP+M+AH0	572.2 ± 10.8	747.6 ± 4.16	835.4 ± 3.51
BP+MCMCS+A	627.6 ± 8.2	779.4 ± 7.77	838 ± 4.12
BP+MCMCS+A0	629 ± 9.82	776 ± 9.14	832.2 ± 7.85
BP+MCMCS+AH	630.4 ± 11.6	$\mathbf{800.8 \pm 8.23}$	881.6 ± 6.19
BP+MCMCS+AH0	$\mathbf{643 \pm 10.5}$	$\mathbf{801.8 \pm 6.65}$	$\mathbf{882 \pm 8.86}$
BP+MCMCW+A	603 ± 30.3	766 ± 6.63	834 ± 1.87
BP+MCMCW+A0	588 ± 36.8	757.6 ± 5.94	834 ± 5.05
BP+MCMCW+AH	$\mathbf{632 \pm 22.9}$	795 ± 14.5	$\mathbf{886.4 \pm 6.84}$
BP+MCMCW+AH0	$\mathbf{635.4 \pm 23}$	$\mathbf{796 \pm 12.1}$	$\mathbf{890.6 \pm 3.21}$

Mean recalls were calculated across five replicate BayesProt runs with differing random seeds. Also shown are estimated standard deviations of recall across the five replicates. The three methods with the highest mean recall for each precision are shown in bold.

For all tested quantification and FDR estimation strategies, the Student's t-test applied to the simple median protein quantifications (BP+T) shows significantly reduced recall across all ranges of precision when compared with the same quantification methods applied to the ProteinPilot (PP+T) and MSstatsTMT (MS+MT) quantification results. Similar but less extreme effects are seen at 1% FDP with the Bayesian t-tests that do not account for protein quantification uncertainty (BP+MCMCS0 and BP+MCMCW0), though they do exhibit some improvement in recall at lower precision (5% and 10%). Improvements over the ProteinPilot and MSstatsTMT results across the range of precisions are made when the uncertainty is taken into account (BP+M, BP+MCMCS, BP+MCMCW). At higher precisions (1%) the reproducibility of the recalls is much lower, as evidenced by the generally higher standard deviations in the 1% column of Table 3.3.

3.4.2 No Peptide Random Effect, Independent Feature Variance

With the addition of per-feature residual variances, but still without a peptide random effect, the model relies entirely on the feature quantifications to infer protein quantifications, downweighting unreliable features' contribution to the overall protein quantification; inference regarding the quality of a peptide's quantification is only inferred indirectly through the feature modelling.

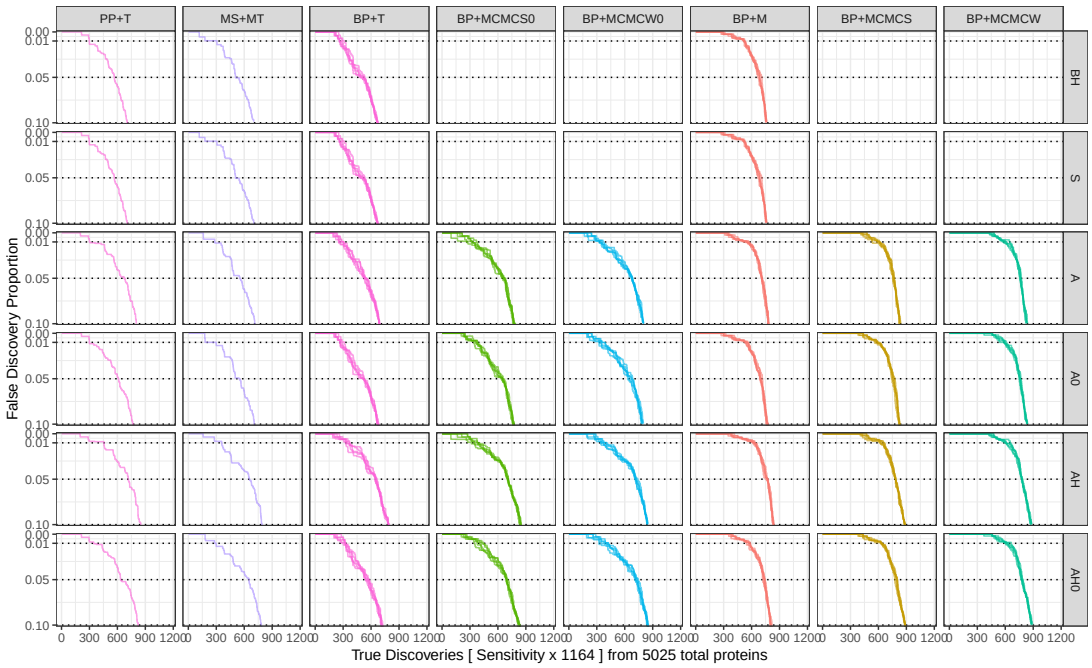

Figure 3.7: Precision–recall curves for single fraction spike-in data with no peptide random effect and per-feature residual variance for BayesProt. Results of repeat runs of BayesProt with differing random seeds are shown as separate lines.

Table 3.4: Mean recalls of tested methods for differential expression at multiple precisions with no peptide modelling and a per-feature residual variance for BayesProt.

	Recall		
Method	1% FDP	5% FDP	10% FDP
PP+T+BH	299	567	707
PP+T+S	299	567	707
PP+T+A	306	671	802
PP+T+A0	300	612	765
PP+T+AH	456	716	851
PP+T+AH0	346	635	823
MS+MT+BH	307	507	702
MS+MT+S	307	507	702
MS+MT+A	283	553	713
MS+MT+A0	299	552	707
MS+MT+AH	358	656	794
MS+MT+AH0	299	651	783
BP+T+BH	275.8 ± 23.5	508.4 ± 20.3	669.2 ± 5.31
BP+T+S	275.8 ± 23.5	508.4 ± 20.3	669.2 ± 5.31
BP+T+A	284 ± 28.9	537.4 ± 11.1	689.4 ± 3.51
BP+T+A0	278.2 ± 20.6	519.2 ± 4.92	675 ± 5.34
BP+T+AH	348.2 ± 17.8	651.4 ± 7.92	788 ± 2.45
BP+T+AH0	319.6 ± 10.2	559 ± 16.2	715.4 ± 6.27
BP+MCMCS0+A	321.2 ± 38.6	663.4 ± 9.29	774.8 ± 9.26
BP+MCMCS0+A0	364.4 ± 15.4	648.2 ± 7.73	767.4 ± 6.19
BP+MCMCS0+AH	364.8 ± 34	707 ± 5.1	842 ± 7.97
BP+MCMCS0+AH0	401.6 ± 38.4	694.2 ± 6.3	828.6 ± 7.83
BP+MCMCW0+A	344.4 ± 17.7	679.2 ± 5.26	799.2 ± 5.54
BP+MCMCW0+A0	341.2 ± 13.7	661.8 ± 21.3	789.4 ± 8.44
BP+MCMCW0+AH	376.6 ± 30.4	721.8 ± 11.1	844.4 ± 6.02
BP+MCMCW0+AH0	386 ± 46.4	725.6 ± 14.6	847 ± 4.12
BP+M+BH	523.4 ± 4.04	687.6 ± 15.4	758.8 ± 3.83
BP+M+S	523.4 ± 4.04	687.6 ± 15.4	758.8 ± 3.83
BP+M+A	542.2 ± 20.6	709.4 ± 2.61	780.8 ± 4.32
BP+M+A0	542.4 ± 13	698.6 ± 10.7	768 ± 7.21
BP+M+AH	$\mathbf{627 \pm 8.66}$	769 ± 5.39	835 ± 3.46
BP+M+AH0	566.6 ± 14	735.2 ± 10.6	807 ± 8.51
BP+MCMCS+A	596.6 ± 35.4	763 ± 8.19	830.8 ± 6.14
BP+MCMCS+A0	585.4 ± 18.9	759.6 ± 12.5	826 ± 4.9
BP+MCMCS+AH	$\mathbf{635.8 \pm 18}$	$\mathbf{784.6 \pm 1.34}$	$\mathbf{883.6 \pm 5.5}$
BP+MCMCS+AH0	$\mathbf{633.4 \pm 4.51}$	$\mathbf{793.4 \pm 11.2}$	$\mathbf{886.4 \pm 2.7}$
BP+MCMCW+A	609.4 ± 26.1	762.8 ± 6.91	830.8 ± 6.53
BP+MCMCW+A0	584.2 ± 26.3	758.2 ± 6.94	834.4 ± 3.36
BP+MCMCW+AH	617 ± 20.7	$\mathbf{780.4 \pm 4.16}$	879.8 ± 4.6
BP+MCMCW+AH0	626.6 ± 24.2	778.2 ± 8.76	$\mathbf{884.6 \pm 5.98}$

Mean recalls were calculated across five replicate BayesProt runs with differing random seeds. Also shown are estimated standard deviations of recall across the five replicates. The three methods with the highest mean recall for each precision are shown in bold.

3.4.3 Single Peptide Variance, Single Feature Variance

In this configuration, a per-sample random effect with a single variance across all peptides is added for each protein.

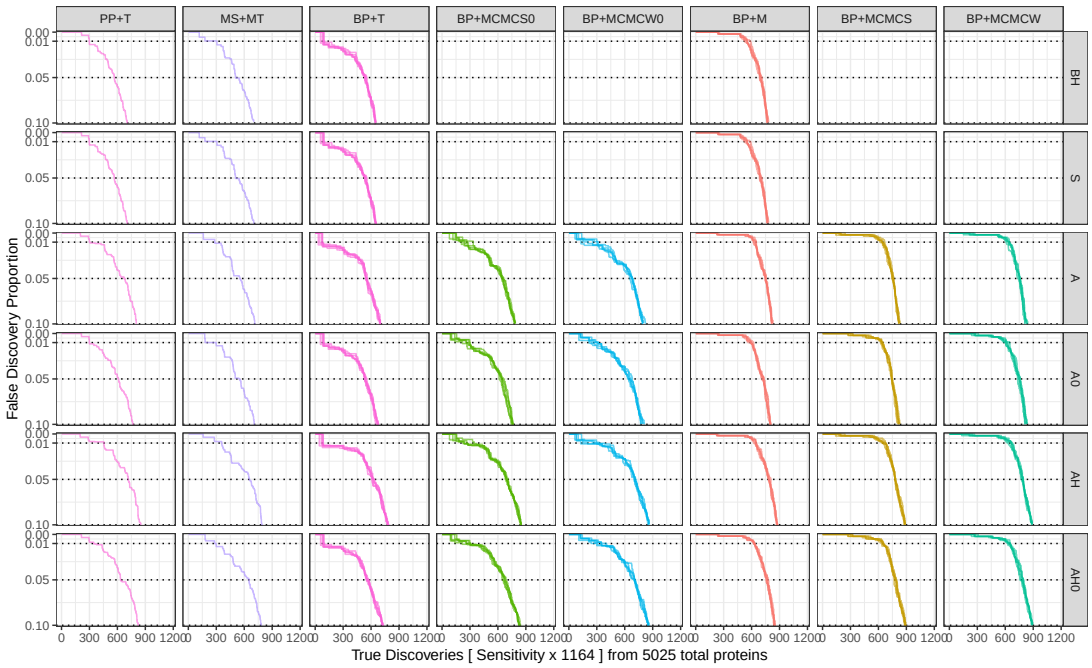

Figure 3.8: Precision-Recall Curves for Spike-in data with BayesProt run with a single per-protein variance across peptide deviations and a single per-protein residual variance. Results of repeat runs of BayesProt with differing random seeds are shown as separate lines.

Table 3.5: Mean recalls of tested methods for differential expression at multiple precisions with a single peptide variance and a single per-protein residual variance for BayesProt.

	Recall		
Method	1% FDP	5% FDP	10% FDP
PP+T+BH	299	567	707
PP+T+S	299	567	707
PP+T+A	306	671	802
PP+T+A0	300	612	765
PP+T+AH	456	716	851
PP+T+AH0	346	635	823
MS+MT+BH	307	507	702
MS+MT+S	307	507	702
MS+MT+A	283	553	713
MS+MT+A0	299	552	707
MS+MT+AH	358	656	794
MS+MT+AH0	299	651	783
BP+T+BH	81.8 ± 16.9	541.4 ± 9.94	650.2 ± 5.26
BP+T+S	81.8 ± 16.9	541.4 ± 9.94	650.2 ± 5.26
BP+T+A	60.6 ± 20	550 ± 9.27	697.8 ± 2.95
BP+T+A0	80 ± 15.3	540.2 ± 9.78	671.2 ± 8.84
BP+T+AH	71.4 ± 17.5	622.4 ± 10.2	780.2 ± 4.97
BP+T+AH0	82 ± 16.3	563.6 ± 11.1	725.2 ± 4.92
BP+MCMCS0+A	255.6 ± 49.4	651.2 ± 7.95	784 ± 3.94
BP+MCMCS0+A0	267.8 ± 34.2	640 ± 19.2	756.6 ± 8.79
BP+MCMCS0+AH	262.2 ± 48.8	694.6 ± 8.91	847.8 ± 1.64
BP+MCMCS0+AH0	278 ± 39.6	670.6 ± 13.9	830.4 ± 7.02
BP+MCMCW0+A	242.2 ± 76.9	674.6 ± 11.5	803 ± 12.7
BP+MCMCW0+A0	278.4 ± 45.6	658 ± 16.4	789 ± 16.6
BP+MCMCW0+AH	241.8 ± 79.2	707.8 ± 5.89	858 ± 6.04
BP+MCMCW0+AH0	341.6 ± 41.4	705.2 ± 5.89	853.8 ± 5.26
BP+M+BH	552.8 ± 18.6	696.4 ± 9.79	773.8 ± 6.42
BP+M+S	552.8 ± 18.6	696.4 ± 9.79	773.8 ± 6.42
BP+M+A	636.6 ± 14.6	751.2 ± 4.6	820.8 ± 6.57
BP+M+A0	596.2 ± 21.6	725.2 ± 8.61	801.4 ± 4.93
BP+M+AH	642 ± 9.59	788.8 ± 10.3	875 ± 5.83
BP+M+AH0	629 ± 5.1	766.4 ± 9.4	847.4 ± 5.98
BP+MCMCS+A	642.6 ± 16.1	755.4 ± 8.29	823.8 ± 6.38
BP+MCMCS+A0	636.8 ± 9.78	745.2 ± 4.76	824.4 ± 4.62
BP+MCMCS+AH	$\mathbf{657.2 \pm 16}$	$\mathbf{781.4 \pm 4.93}$	886.6 ± 3.58
BP+MCMCS+AH0	$\mathbf{646.6 \pm 7.64}$	$\mathbf{782.6 \pm 9.61}$	$\mathbf{887.8 \pm 5.72}$
BP+MCMCW+A	633.2 ± 13.2	752.2 ± 7.29	825 ± 12.9
BP+MCMCW+A0	618.2 ± 17.2	749.8 ± 12.6	826.2 ± 9.86
BP+MCMCW+AH	$\mathbf{657.2 \pm 15.6}$	$\mathbf{785.6 \pm 9.71}$	$\mathbf{887 \pm 3.87}$
BP+MCMCW+AH0	641.4 ± 11	780.8 ± 14.1	$\mathbf{892.4 \pm 2.7}$

Mean recalls were calculated across five replicate BayesProt runs with differing random seeds. Also shown are estimated standard deviations of recall across the five replicates. The three methods with the highest mean recall for each precision are shown in bold.

3.4.4 Single Peptide Variance, Independent Feature Variance

Here, the independent residual variances for each feature mean that "unreliable" features can have their contribution to the overall protein quantification down-weighted.

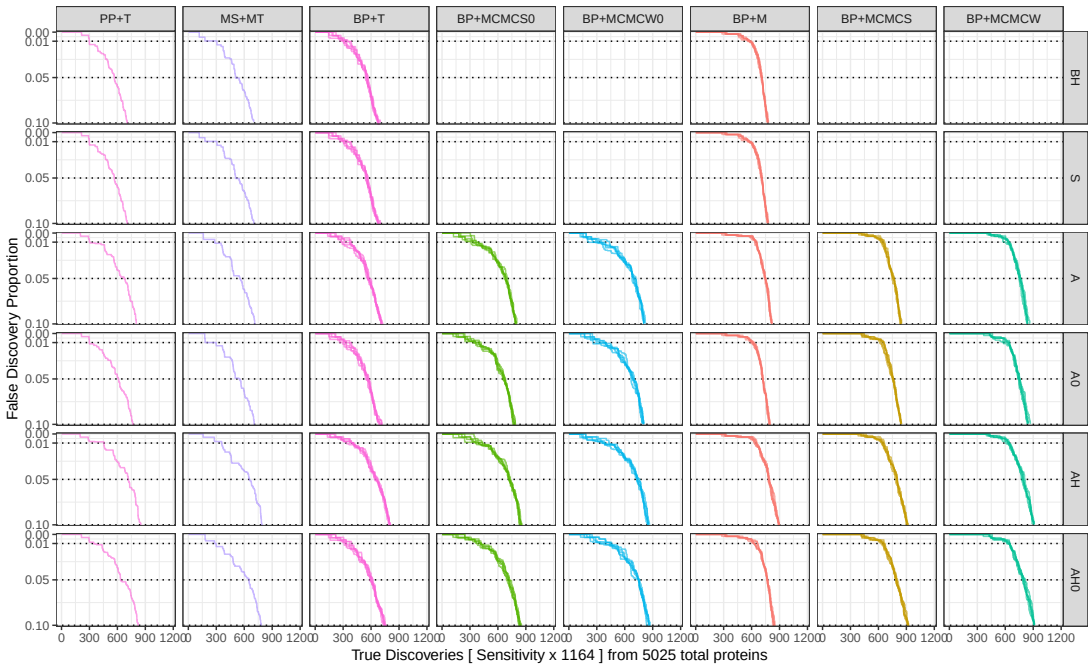

Figure 3.9: Precision–recall curves for single fraction spike-in data with a single per-protein variance across peptide deviations and independent, per-feature residual variances for BayesProt. Results of repeat runs of BayesProt with differing random seeds are shown as separate lines.

The per-feature residual variance allows the model to "down-weight" features whose quantification pattern is inconsistent with the overall consensus across the protein, reducing the contribution of these unreliable features to the subsequent peptide and protein quantifications.

Table 3.6: Mean recalls of tested methods for differential expression at multiple precisions with a single peptide variance and a per-feature residual variance for BayesProt.

	Recall		
Method	1% FDP	5% FDP	10% FDP
PP+T+BH	299	567	707
PP+T+S	299	567	707
PP+T+A	306	671	802
PP+T+A0	300	612	765
PP+T+AH	456	716	851
PP+T+AH0	346	635	823
MS+MT+BH	307	507	702
MS+MT+S	307	507	702
MS+MT+A	283	553	713
MS+MT+A0	299	552	707
MS+MT+AH	358	656	794
MS+MT+AH0	299	651	783
BP+T+BH	324 ± 18.5	555.2 ± 10.9	683.4 ± 13
BP+T+S	324 ± 18.5	555.2 ± 10.9	683.4 ± 13
BP+T+A	330.2 ± 45.1	569 ± 8.19	713.4 ± 7.23
BP+T+A0	329 ± 27.7	561.6 ± 13.3	702.8 ± 12.4
BP+T+AH	396 ± 32.6	658.6 ± 13.2	800.4 ± 8.17
BP+T+AH0	346.6 ± 31.8	604.4 ± 12.4	747.4 ± 7.8
BP+MCMCS0+A	341.4 ± 22.1	685.6 ± 13.4	795.4 ± 5.86
BP+MCMCS0+A0	380.8 ± 35.7	667.4 ± 10.7	777.8 ± 11.6
BP+MCMCS0+AH	408.4 ± 29.5	722.8 ± 7.19	845.8 ± 9.83
BP+MCMCS0+AH0	408.2 ± 21	712.4 ± 14	837.6 ± 11.3
BP+MCMCW0+A	312.8 ± 86.4	699 ± 15	809.4 ± 9.45
BP+MCMCW0+A0	343.4 ± 64.3	688.2 ± 13.7	799.6 ± 6.77
BP+MCMCW0+AH	387.2 ± 29.5	741.4 ± 13.4	856.2 ± 6.87
BP+MCMCW0+AH0	416.8 ± 46.8	751.8 ± 6.65	858.2 ± 13.1
BP+M+BH	583.8 ± 19	709.2 ± 4.09	775.8 ± 4.32
BP+M+S	583.8 ± 19	709.2 ± 4.09	775.8 ± 4.32
BP+M+A	$\mathbf{645.6 \pm 8.56}$	748.4 ± 3.58	816.8 ± 5.26
BP+M+A0	613.2 ± 12.4	720.2 ± 3.19	794.6 ± 3.58
BP+M+AH	644.8 ± 12.3	784.8 ± 5.36	893.4 ± 4.77
BP+M+AH0	642.6 ± 19.3	768.6 ± 2.97	841 ± 8.28
BP+MCMCS+A	645 ± 14.8	761.4 ± 8.96	845.4 ± 3.21
BP+MCMCS+A0	642.6 ± 18.9	757 ± 6.44	844.4 ± 3.91
BP+MCMCS+AH	$\mathbf{647.2 \pm 10.5}$	$\mathbf{797.6 \pm 7.3}$	$\mathbf{909.6 \pm 6.5}$
BP+MCMCS+AH0	$\mathbf{650 \pm 12.3}$	$\mathbf{793.6 \pm 6.99}$	$\mathbf{918.6 \pm 5.27}$
BP+MCMCW+A	629 ± 14.6	758.8 ± 7.82	844.2 ± 12.6
BP+MCMCW+A0	624.8 ± 27.5	748.2 ± 6.38	847.8 ± 12.6
BP+MCMCW+AH	632 ± 13.5	792.2 ± 11.4	904.6 ± 9.18
BP+MCMCW+AH0	637.8 ± 11.5	$\mathbf{798 \pm 10.7}$	$\mathbf{911.4 \pm 5.59}$

Mean recalls were calculated across five replicate BayesProt runs with differing random seeds. Also shown are estimated standard deviations of recall across the five replicates. The three methods with the highest mean recall for each precision are shown in bold.

3.4.5 Independent Peptide Variance, Single Feature Variance

Switching to a per-peptide variance across the peptide deviations means that unreliable, more variable peptides can be down-weighted in the model.

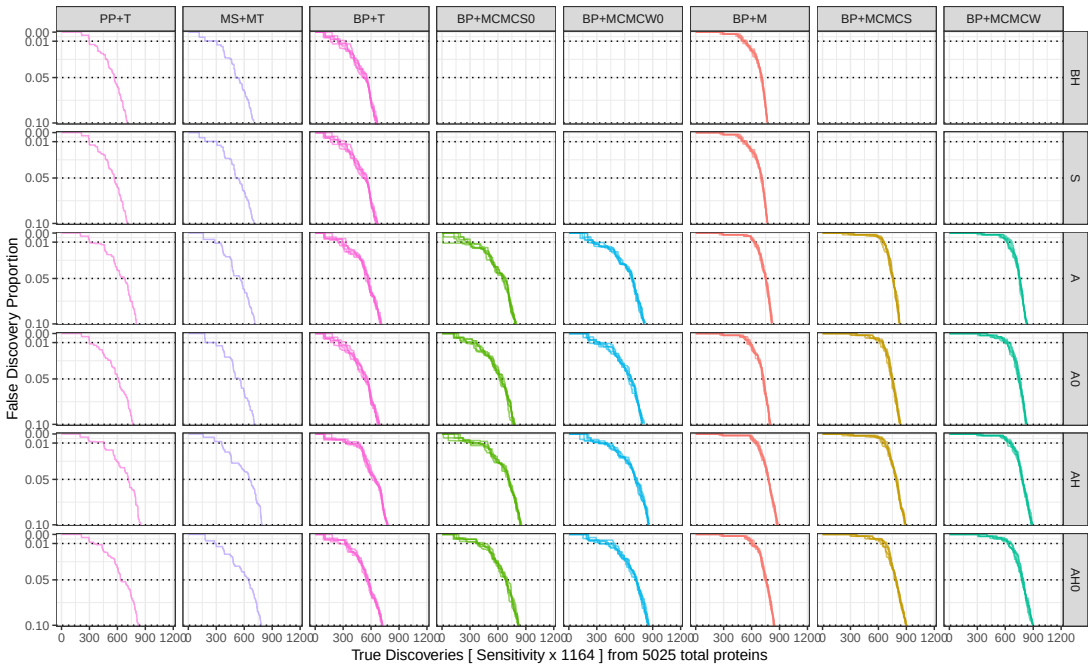

Figure 3.10: Precision–recall curves for single fraction spike-in data with independent per-peptide variance across peptide deviations and a single per-protein residual variance for BayesProt. Results of repeat runs of BayesProt with differing random seeds are shown as separate lines.

Table 3.7: Mean recalls of tested methods for differential expression at multiple precisions with per-peptide variances across peptide deviations and per-protein residual variances for BayesProt.

	Recall		
Method	1% FDP	5% FDP	10% FDP
PP+T+BH	299	567	707
PP+T+S	299	567	707
PP+T+A	306	671	802
PP+T+A0	300	612	765
PP+T+AH	456	716	851
PP+T+AH0	346	635	823
MS+MT+BH	307	507	702
MS+MT+S	307	507	702
MS+MT+A	283	553	713
MS+MT+A0	299	552	707
MS+MT+AH	358	656	794
MS+MT+AH0	299	651	783
BP+T+BH	244.4 ± 56.3	541.4 ± 11.1	662.8 ± 8.01
BP+T+S	244.4 ± 56.3	541.4 ± 11.1	662.8 ± 8.01
BP+T+A	286.6 ± 38.5	561.8 ± 13.5	704.6 ± 3.36
BP+T+A0	257.6 ± 45.8	548.4 ± 14.3	684 ± 6.44
BP+T+AH	361.8 ± 57.9	638.8 ± 12.9	775.6 ± 4.22
BP+T+AH0	315 ± 39.4	560.8 ± 13.4	720.4 ± 4.34
BP+MCMCS0+A	222.6 ± 124	668.2 ± 17.1	791.4 ± 7.44
BP+MCMCS0+A0	324.8 ± 46.1	637 ± 18.9	770.8 ± 13
BP+MCMCS0+AH	384.6 ± 76.7	718.2 ± 6.06	844.2 ± 6.3
BP+MCMCS0+AH0	398.8 ± 29.7	688.8 ± 8.11	820.4 ± 4.72
BP+MCMCW0+A	245.2 ± 21.7	680.2 ± 6.65	811.6 ± 6.88
BP+MCMCW0+A0	292 ± 45.6	660.6 ± 13.9	794.4 ± 11.1
BP+MCMCW0+AH	359 ± 49.1	720 ± 12.1	858 ± 6.04
BP+MCMCW0+AH0	429 ± 64.4	723.8 ± 7.19	854 ± 4.85
BP+M+BH	528.4 ± 16.7	705.8 ± 6.76	769.2 ± 3.7
BP+M+S	528.4 ± 16.7	705.8 ± 6.76	769.2 ± 3.7
BP+M+A	632.4 ± 6.39	752 ± 3.61	819.4 ± 4.72
BP+M+A0	569 ± 18.7	726 ± 5.34	798 ± 6.78
BP+M+AH	$\mathbf{653.2 \pm 15.8}$	783.4 ± 4.39	877 ± 5.57
BP+M+AH0	610.8 ± 12.8	751 ± 8.86	840.8 ± 5.97
BP+MCMCS+A	651.4 ± 12.6	756.4 ± 3.85	828.8 ± 3.19
BP+MCMCS+A0	638 ± 17.9	749.8 ± 7.16	834 ± 3.08
BP+MCMCS+AH	$\mathbf{663.8 \pm 24.8}$	$\mathbf{793 \pm 5.52}$	889.6 ± 3.78
BP+MCMCS+AH0	$\mathbf{659.6 \pm 28}$	$\mathbf{785.2 \pm 5.72}$	$\mathbf{895.8 \pm 2.39}$
BP+MCMCW+A	639 ± 22.8	753.2 ± 3.7	830.6 ± 5.03
BP+MCMCW+A0	618 ± 15.7	748.8 ± 10.6	826.2 ± 6.34
BP+MCMCW+AH	650 ± 17.4	783.4 ± 4.83	$\mathbf{890 \pm 9}$
BP+MCMCW+AH0	642.4 ± 17.2	$\mathbf{786.4 \pm 8.11}$	$\mathbf{895.4 \pm 4.39}$

Mean recalls were calculated across five replicate BayesProt runs with differing random seeds. Also shown are estimated standard deviations of recall across the five replicates. The three methods with the highest mean recall for each precision are shown in bold.

3.4.6 Default BayesProt Parameters — Independent Peptide Variance, Independent Feature Variance

The default configuration of BayesProt is a model with per-peptide variances across peptide deviations and per-feature residual variances.

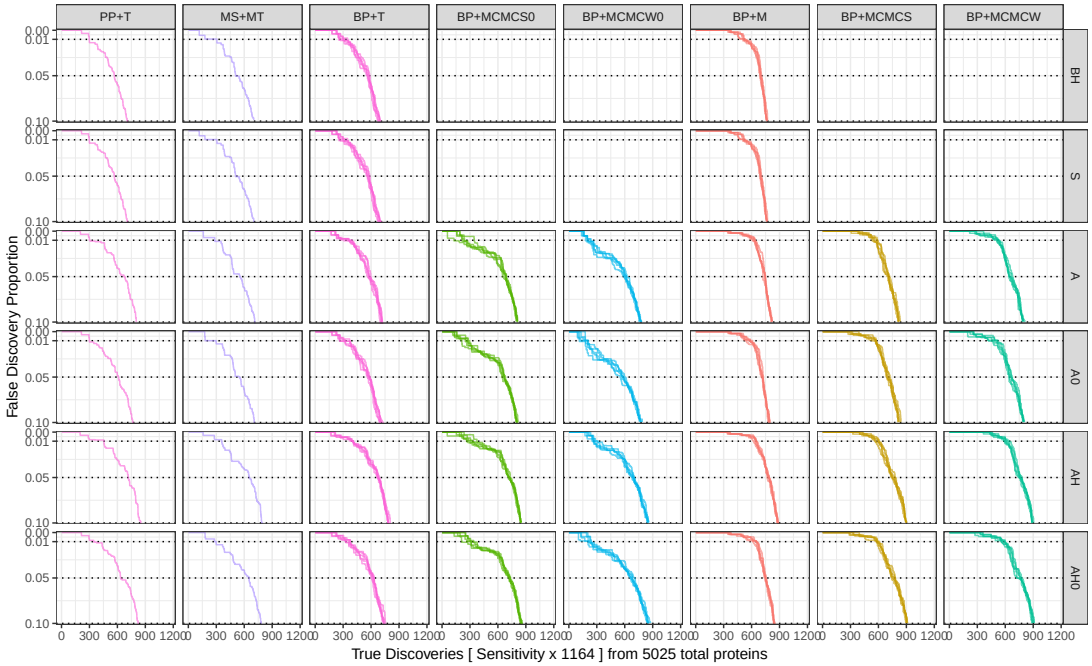

Figure 3.11: Precision-Recall Curves for Spike-in data. Results of repeat runs of BayesProt with differing random seeds are shown as separate lines.

Table 3.8: Mean recalls of tested methods for differential expression at multiple precisions with per-peptide variances across peptide deviations and per-feature residual variances for BayesProt.

	Recall		
Method	1% FDP	5% FDP	10% FDP
PP+T+BH	299	567	707
PP+T+S	299	567	707
PP+T+A	306	671	802
PP+T+A0	300	612	765
PP+T+AH	456	716	851
PP+T+AH0	346	635	823
MS+MT+BH	307	507	702
MS+MT+S	307	507	702
MS+MT+A	283	553	713
MS+MT+A0	299	552	707
MS+MT+AH	358	656	794
MS+MT+AH0	299	651	783
BP+T+BH	297.6 ± 38.8	570.6 ± 11	696 ± 12.9
BP+T+S	297.6 ± 38.8	570.6 ± 11	696 ± 12.9
BP+T+A	356.8 ± 33.6	578.2 ± 12.8	712.8 ± 10.9
BP+T+A0	307.2 ± 45.1	575.8 ± 10.3	704.6 ± 15.1
BP+T+AH	409.2 ± 2.39	676.8 ± 8.35	790.4 ± 8.73
BP+T+AH0	354 ± 23.8	621.2 ± 13.1	740.4 ± 9.53
BP+MCMCS0+A	232.8 ± 77.7	680.2 ± 15.5	808 ± 6.2
BP+MCMCS0+A0	258.6 ± 31.2	666.8 ± 11.6	806.2 ± 7.36
BP+MCMCS0+AH	317.4 ± 40.6	722.8 ± 7.79	848.8 ± 4.09
BP+MCMCS0+AH0	296.8 ± 48.8	726.2 ± 9.63	854.4 ± 5.68
BP+MCMCW0+A	210.8 ± 23.2	609.8 ± 12.4	770.8 ± 10.9
BP+MCMCW0+A0	180.6 ± 39.8	577.8 ± 16.8	771.6 ± 11.9
BP+MCMCW0+AH	282.8 ± 22.7	690.8 ± 8.84	852.2 ± 7.16
BP+MCMCW0+AH0	223.2 ± 14.1	679.2 ± 20.6	859.4 ± 10.7
BP+M+BH	520 ± 9.3	697 ± 7.81	767.4 ± 3.91
BP+M+S	520 ± 9.3	697 ± 7.81	767.4 ± 3.91
BP+M+A	$\mathbf{627.4 \pm 9.66}$	741.6 ± 11.2	818.2 ± 4.21
BP+M+A0	576.8 ± 26.9	716.2 ± 7.95	789.8 ± 6.02
BP+M+AH	$\mathbf{642.2 \pm 11.2}$	$\mathbf{782.4 \pm 9.56}$	883.4 ± 2.41
BP+M+AH0	$\mathbf{643.2 \pm 15.5}$	750.4 ± 4.04	842.4 ± 5.94
BP+MCMCS+A	563.2 ± 25.4	708.8 ± 8.58	820.2 ± 10
BP+MCMCS+A0	562.8 ± 17.5	703 ± 14.1	822.4 ± 13.6
BP+MCMCS+AH	588.6 ± 23.2	$\mathbf{760.8 \pm 18.6}$	$\mathbf{896.2 \pm 5.54}$
BP+MCMCS+AH0	592.6 ± 14.2	$\mathbf{766.6 \pm 16.4}$	$\mathbf{904.6 \pm 6.58}$
BP+MCMCW+A	522.2 ± 9.73	656.4 ± 16.4	793.4 ± 12.9
BP+MCMCW+A0	484.8 ± 41.7	658 ± 14.6	799.6 ± 6.58
BP+MCMCW+AH	612.8 ± 18.9	758 ± 13.7	893.6 ± 6.62
BP+MCMCW+AH0	581.6 ± 22.2	753 ± 25.2	$\mathbf{899.6 \pm 8.85}$

Mean recalls were calculated across five replicate BayesProt runs with differing random seeds. Also shown are estimated standard deviations of recall across the five replicates. The three methods with the highest mean recall for each precision are shown in bold.

3.4.7 Full Empirical Bayes

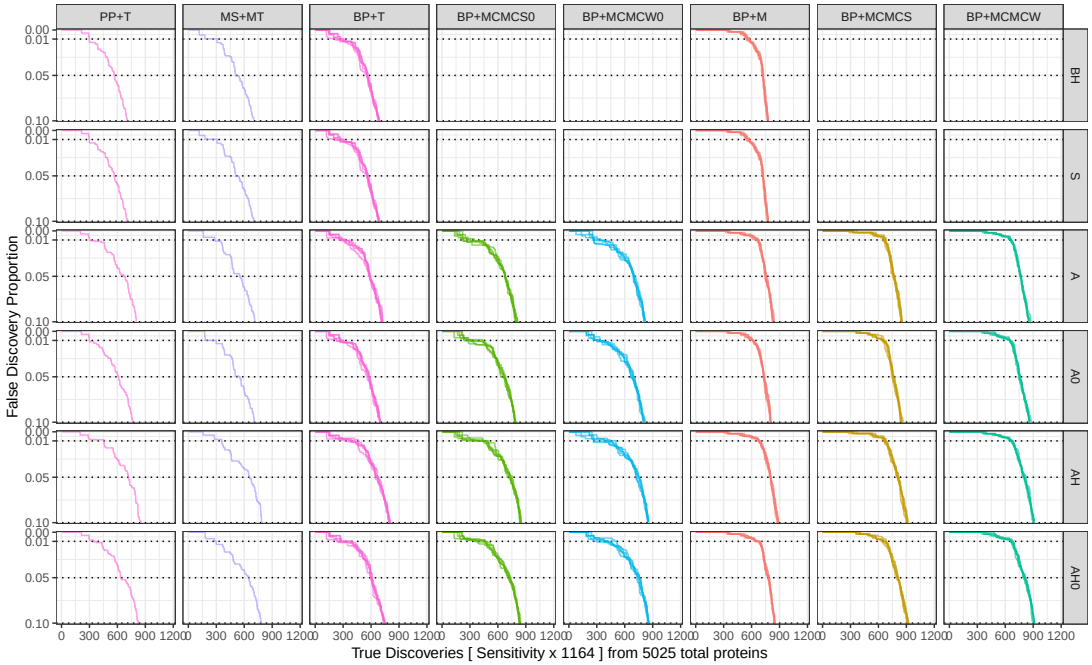

Figure 3.12: Precision-Recall curves for single fraction spike-in data with independent per-peptide variances across peptide deviations and independent per-feature residual variances for BayesProt with a full empirical Bayes method with all proteins using the informative priors on peptide and residual variances. Results of repeat runs of BayesProt with differing random seeds are shown as separate lines.

Table 3.9: Mean recalls of tested methods for differential expression at multiple precisions with full empirical Bayes model.

	Recall		
Method	1% FDP	5% FDP	10% FDP
PP+T+BH	299	567	707
PP+T+S	299	567	707
PP+T+A	306	671	802
PP+T+A0	300	612	765
PP+T+AH	456	716	851
PP+T+AH0	346	635	823
MS+MT+BH	307	507	702
MS+MT+S	307	507	702
MS+MT+A	283	553	713
MS+MT+A0	299	552	707
MS+MT+AH	358	656	794
MS+MT+AH0	299	651	783
BP+T+BH	264.6 ± 40.9	561 ± 4.9	684.2 ± 6.38
BP+T+S	264.6 ± 40.9	561 ± 4.9	684.2 ± 6.38
BP+T+A	297.2 ± 29.4	589.4 ± 5.22	721 ± 4.85
BP+T+A0	265.6 ± 36.2	574 ± 11.8	696.6 ± 1.82
BP+T+AH	401.4 ± 51.3	661.8 ± 13.1	798.4 ± 5.94
BP+T+AH0	289.8 ± 67.6	603 ± 7.97	743.4 ± 7.5
BP+MCMCS0+A	274.6 ± 52.1	676.2 ± 1.1	805 ± 6.86
BP+MCMCS0+A0	319.2 ± 73.2	661.2 ± 17.8	789.2 ± 6.98
BP+MCMCS0+AH	403.8 ± 87.6	732.2 ± 13.8	843.8 ± 8.7
BP+MCMCS0+AH0	423.4 ± 45.2	722.8 ± 8.23	836.6 ± 7.92
BP+MCMCW0+A	290.2 ± 37.1	695.2 ± 3.9	813.6 ± 6.15
BP+MCMCW0+A0	318.6 ± 16.1	682.4 ± 17	808.6 ± 9.15
BP+MCMCW0+AH	377.2 ± 107	740.4 ± 14.8	853.4 ± 4.72
BP+MCMCW0+AH0	406.4 ± 76	749.6 ± 12.3	857.6 ± 6.8
BP+M+BH	559.6 ± 15.9	720.4 ± 6.19	776.4 ± 2.61
BP+M+S	559.6 ± 15.9	720.4 ± 6.19	776.4 ± 2.61
BP+M+A	658 ± 12.6	753 ± 6.86	834.4 ± 6.11
BP+M+A0	614.8 ± 3.56	733.2 ± 6.61	805.4 ± 3.85
BP+M+AH	$\mathbf{685.4 \pm 10.7}$	803.8 ± 3.35	882.4 ± 6.07
BP+M+AH0	659.6 ± 24	784.4 ± 6.58	847.8 ± 4.55
BP+MCMCS+A	665.8 ± 12.6	767 ± 9.43	852 ± 8.09
BP+MCMCS+A0	650.6 ± 8.2	757.8 ± 8.01	848.4 ± 11.9
BP+MCMCS+AH	$\mathbf{669.8 \pm 10.9}$	$\mathbf{810.4 \pm 8.02}$	$\mathbf{911 \pm 8.8}$
BP+MCMCS+AH0	657.4 ± 17	805.8 ± 8.76	$\mathbf{917.8 \pm 6.26}$
BP+MCMCW+A	654.4 ± 7.64	766.6 ± 4.62	860.4 ± 9.84
BP+MCMCW+A0	648 ± 2.74	754 ± 7.38	863.8 ± 8.26
BP+MCMCW+AH	659.4 ± 10.3	$\mathbf{807.2 \pm 10.4}$	906 ± 3.81
BP+MCMCW+AH0	$\mathbf{666.8 \pm 10.5}$	$\mathbf{807.6 \pm 5.55}$	$\mathbf{913.6 \pm 3.91}$

Mean recalls were calculated across five replicate BayesProt runs with differing random seeds. Also shown are estimated standard deviations of recall across the five replicates. The three methods with the highest mean recall for each precision are shown in bold.

3.4.8 Discussion of Precision-Recall Results

From the above results a number of broad conclusions can be made.

Uncertainty Propagation

The above results go some way to demonstrate the importance of the propagation of uncertainty: simply taking a simple average statistic (in this case, the median values) fails to account for the uncertainty in each protein quantification. Proper handling of this uncertainty through the use of the meta-analytic t-test and MCMC-based Bayesian t-test leads to better recall across a range of precision thresholds, implying an improved ranking of proteins.

Bayesian T-Tests

At higher precisions, the unequal-variances Bayesian t-test exhibits a slight reduction in power over the equal-variances t-test; on the other hand, BP+MCMCS+AH and BP+MCMCS+AH0 consistently rank among the top three methods at 1%, 5% and 10% FDP.

Differences in the Configuration of the ash R Package

Additionally, it is evident from comparing results for the same quantification method with differing configurations of ash in the above tables that the choice of a half-uniform distribution for ash yields an increase in power over the uniform distribution; for example, for the default BayesProt configuration with the full empirical Bayes method in Table 3.9, BP+MCMCS+A0 achieves a mean recall of 650.6, 757.8 and 848.4 at 1%, 5% and 10% respectively. The corresponding method with a half-uniform distribution, BP+MCMCS+AH0 achieves higher mean recalls of 657.4, 805.8 and 917.8. By allowing for a non-symmetric distribution of fold-changes across the population, the power to detect true positives is increased.

The effect of the chosen value for the degrees of freedom parameter for ash is less obvious. For the metafor t-test, setting *df* to infinity (A0 and AH0), meaning that the quantification estimates are assumed to be normally distributed, results in a decrease in recall over estimating the *df* parameter separately for each protein; for example, comparing BP+M+A0 with BP+M+A and BP+M+AH0 with BP+M+AH in each of the above tables shows this. This difference can be likened to the difference between Z-statistics and T-statistics; allowing for more heavily tailed quantification estimates

increases the power of the metafor t-test. The same effect on the Bayesian t-test is not readily apparent, with much smaller differences in mean recall, often in the opposite direction.

The differences (and therefore advantages or disadvantages) resulting from specifying a degrees of freedom parameter for ash when applied to the Bayesian t-test are not obvious from the above results, but will become clearer in Section (3.5) when the estimated FDRs are compared with the ground-truth FDPs.

Choice of BayesProt Configuration

Comparing the results of one of the methods across all of the tested BayesProt configurations gives a broad picture of how the choice of peptide and feature modelling affects the ordering of proteins. Table 3.10 shows the results of BP+MCMCS+AH across all of the tested BayesProt configurations above.

Bayesprot Configuration		Recall		
Peptide Variance	Residual Variance	1% FDP	5% FDP	10% FDP
None	Per-Protein	630.4 ± 11.6	800.8 ± 8.23	881.6 ± 6.19
None	Per-Feature	635.8 ± 18	784.6 ± 1.34	883.6 ± 5.5
Per-Protein	Per-Protein	657.2 ± 16	781.4 ± 4.93	886.6 ± 3.58
Per-Protein	Per-Feature	647.2 ± 10.5	797.6 ± 7.3	909.6 ± 6.5
Per-Peptide	Per-Protein	663.8 ± 24.8	793 ± 5.52	889.6 ± 3.78
Per-Peptide	Per-Feature	588.6 ± 23.2	760.8 ± 18.6	896.2 ± 5.54
Per-Peptide (EB)	Per-Feature (EB)	$\mathbf{669.8 \pm 10.9}$	$\mathbf{810.4 \pm 8.02}$	$\mathbf{911 \pm 8.8}$

Table 3.10: Mean recall with standard deviation of the equal-variance Bayesian t-test across the tested configurations of the BayesProt model. FDR is estimated by ash with a half-uniform mixture distribution and *df* estimated separately for each protein. Means and standard deviations of recall are calculated across replicated BayesProt runs.

Ignoring the full empirical Bayes method (EB) for the time being, for this highlighted quantification and FDR estimation method, the default BayesProt settings of per-peptide peptide variance and per-feature residual variances exhibits a reduced mean recall over the other methods at both 1% and 5% FDP, but is only bettered by the single peptide-variance, per-feature residual variance configuration at 10% FDP.

This significant reduction in recall for the default BayesProt configuration at higher precisions is likely due to the increased number of parameters in the model, meaning that the probability mass is spread out over a higher-dimensional parameter space,

resulting in inflated uncertainty of the protein quantification estimates.

Table 3.10 suggests that the choice of a per-feature residual variance over a per-protein residual variance results in a significant decrease in recall across the range of precisions. Hence, there is evidence that this slightly simpler model should be chosen over the more complex model with per-feature residual variances.

There is an argument that the model with a per-protein peptide variance and a per-feature residual variance should be chosen as the default, since for the single fraction spike-in data set (see Table 3.10) this configuration achieves higher recalls at 5% and 10% FDP over the default configuration of BayesProt. However, this model would not allow for post-hoc inspection and analysis of peptide-level quantifications, which would be of interest in scenarios where the presence of proteoforms that have differing profiles of differential expression would result in peptide quantifications that differ from the protein-level quantification.

The reduction in recall that is exhibited by the default configuration of BayesProt is mitigated by the full empirical Bayes procedure; variation in protein quantification uncertainty is regularised, leading to increased recall at all precisions over all of the other tested configurations. In Table 3.10, using the results of the Bayesian equal-variances t-test to make comparisons across configurations of BayesProt, it can be seen that the full empirical Bayes method achieves the highest recall at 1%, 5% and 10% FDP.

The simple t-test on median point estimate protein quantifications shows a general decrease in recall when the full empirical Bayes is applied. Meanwhile, the Bayesian t-tests without uncertainty show some improvement of recall at the higher precisions, though still fail to match the performance of the methods which incorporate uncertainty.

The metafor t-test shows a modest improvement in mean recall with the full empirical Bayes method. For example, at 1% FDP, BP+M+AH previously showed a mean recall of 642.2, rising to 685.4; at 10% FDP, no improvement was shown (882.4 versus 883.4). In this instance it maintains the highest ranked recall at 1% FDP.

False Discovery Rate

In a real proteomics experiment, the true differentially expressed proteins are unknown; comparing methods solely on their ability to provide good ordering of proteins in a data set ignores the fact that practitioners are dependent on estimates of false discovery rate to select sets of proteins for downstream analysis. The following section, Section 3.5, now considers the reported FDRs of each of the methods, comparing them with the

ground-truth FDPs.

3.5 False Discovery Rate Estimation

As discussed in Chapter 2, the FDR provides a measure of the proportion of false positives in a given set of discoveries. Ideally, the FDR reported by a given method would accurately predict the true FDP at any given point in the ordered results. Two additional data sets were used for validating the methods for FDR estimation.

The same spike-in experiment as in Section 3.4 was also run with a different, experimental fractionation method; rather than individual fractions being run separately, sets of three fractions were pooled together. Though this ultimately did not give superior results, it enables the robustness of these differential expression methods in the presence of more noisy data to be evaluated.

The spike-in experiment as it was originally run resulted in some poor quality data, possibly due to some incomplete digestion or pipetting of a number of samples. This prompted a repeat of the experiment. However, this faulty data still proves useful since the robustness of the quantification and differential expression testing in the presence of poor quality data can be analysed.

Firstly, Precision-Recall curves are again presented for each combination of quantification methods (t-test, Metafor t-test, Bayesian t-tests) and methods for FDR estimation (BH, Storey, ash) as applied to protein quantification estimates generated by BayesProt in its default configuration. Alongside these PR curves, curves showing showing the predicted FDR in place of FDP are plotted as dashed lines in Figures 3.13, 3.15 and 3.17. As above, results from replicate runs of BayesProt are shown as separate curves.

To better illustrate the FDR calibration, curves showing the FDP against FDR for FDR/FDP < 0.1 are presented for the above data sets in Figures 3.14, 3.16 and 3.18. An ideal method would give a perfect diagonal line where FDR = FDP, which is plotted with a dotted line. Here, methods whose FDP/FDR curve lie below the diagonal line can be considered to be conservative. Above the diagonal, methods are anti-conservative and FDR control is lost.

Finally, a table of "calibrated recall" values is presented for each of the three data sets in Tables 3.11, 3.12, and 3.13 respectively. This calibrated recall value is calculated as the mean recall at an FDR cutoff where the ground-truth FDP at that point is less than the reported FDR. In a real study, where ground-truth FDP values are not generally observable, practitioners rely on the FDR to be well-calibrated to ensure that

a candidate list of proteins has a limited proportion of false positives. Methods which give overly conservative FDR estimates are unlikely to be adopted, since researchers are likely to resort to methods which give the desired result — a longer list of significant proteins — by, for example, simply adopting a p-value cutoff of 5%.

3.5.1 Single Fraction Spike-in Data

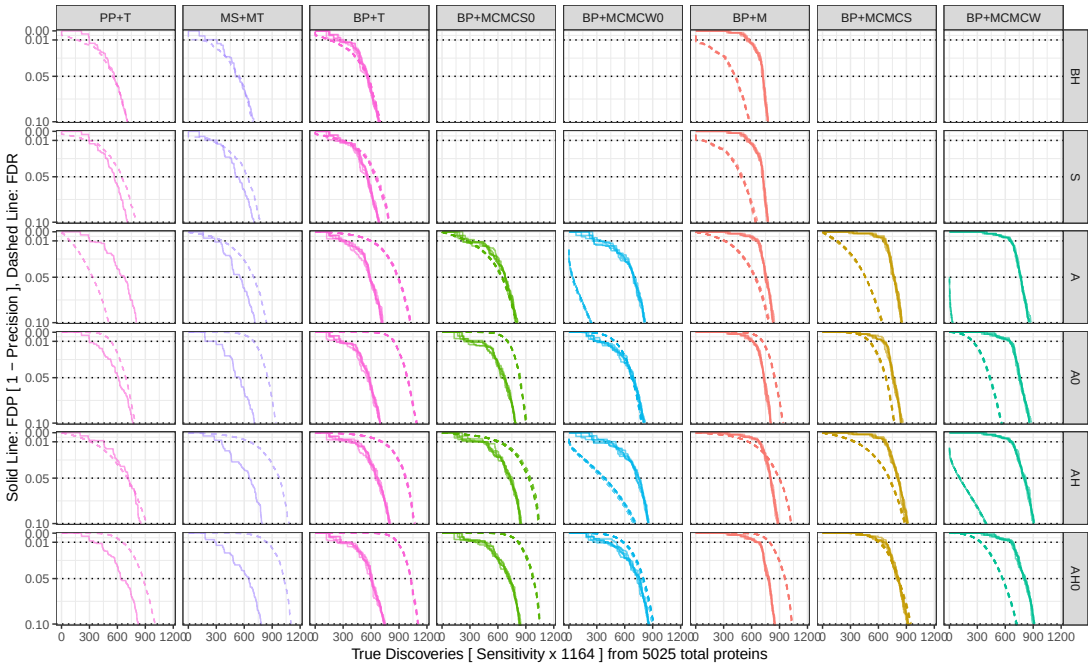

Figure 3.13: Precision-recall curves with FDR for single fraction spike-in data. Results of repeat runs of BayesProt with differing random seeds are shown as separate lines. FDP is shown as solid lines, whereas FDR is shown with dashed lines.

While not giving recall as good as the other methods, applying standard t-tests to the ProteinPilot quants and median BayesProt quants and MSstatsTMT's moderated t-test in combination with Benjamini–Hochberg correction do give calibrated estimates of FDR. Applying the ash methods to these p-values, however, results in anti-conservative FDR estimates; without the additional information concerning effect uncertainty, ash struggles to control the FDR. Conversely, the meta-analytic t-test in combination with Benjamini–Hochberg correction and Storey's q-value method gives conservative FDR estimates. The FDR estimates generated by ash in combination with the meta-analytic t-test are not consistently calibrated. As noted above, the MCMC

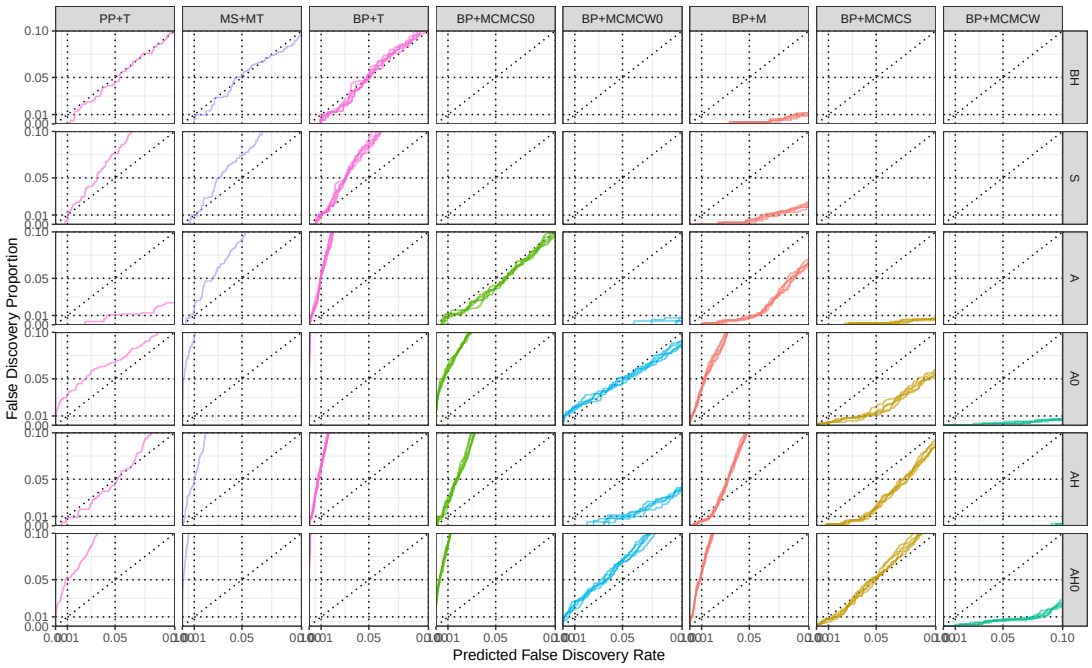

Figure 3.14: FDP vs FDR curves for the single fraction spike-in data. Results of repeat runs of BayesProt with differing random seeds are shown as separate lines.

Table 3.11: Calibrated mean recalls of tested methods for single fraction spike-in data.

	Calibrated Recall		
Method	1% FDR	5% FDR	10% FDR
PP+T+BH	170	546	
PP+T+S	290		
PP+T+A	92	335	508
PP+T+A0			
PP+T+AH	311	708	
PP+T+AH0			
MS+MT+BH	119		690
MS+MT+S	257		
MS+MT+A			
MS+MT+A0			
MS+MT+AH			
MS+MT+AH0			
BP+T+BH	150.8 ± 7.98		
BP+T+S			
BP+T+A			
BP+T+A0			
BP+T+AH			
BP+T+AH0			
BP+MCMCS0+A		$\mathbf{643.2 \pm 5.31}$	
BP+MCMCS0+A0			
BP+MCMCS0+AH			
BP+MCMCS0+AH0			
BP+MCMCW0+A		40.4 ± 4.72	239 ± 5.83
BP+MCMCW0+A0			$\mathbf{783.6 \pm 4.16}$
BP+MCMCW0+AH	2.4 ± 1.14	377.6 ± 12.1	712 ± 12.2
BP+MCMCW0+AH0			
BP+M+BH	1 ± 0	424.8 ± 4.32	575 ± 3.94
BP+M+S	60.2 ± 34.6	493.2 ± 11	647.2 ± 12.2
BP+M+A	309.8 ± 5.85	637.6 ± 6.54	$\mathbf{791.2 \pm 5.76}$
BP+M+A0			
BP+M+AH	$\mathbf{565.4 \pm 1.14}$		
BP+M+AH0			
BP+MCMCS+A	153.2 ± 9.15	461 ± 2.12	641.2 ± 4.76
BP+MCMCS+A0	$\mathbf{501.8 \pm 4.02}$	$\mathbf{680 \pm 3.87}$	772 ± 5.39
BP+MCMCS+AH	346.4 ± 4.28	$\mathbf{714.6 \pm 4.62}$	$\mathbf{888.8 \pm 6.57}$
BP+MCMCS+AH0	$\mathbf{607.8 \pm 6.69}$		
BP+MCMCW+A			33.8 ± 2.68
BP+MCMCW+A0	249.8 ± 3.56	435 ± 4.42	560 ± 3.39
BP+MCMCW+AH		110.6 ± 3.36	394.6 ± 4.93
BP+MCMCW+AH0	337.6 ± 4.04	571.2 ± 2.59	722.6 ± 5.5

Mean values at FDR cutoffs of 1%, 5% and 10% calculated where all replicated BayesProt runs exhibit calibrated FDR at that particular FDR cutoff.

tests which incorporate uncertainty achieve the best recall of proteins. However, in a similar result to the meta-analytic t-test, the FDR estimates generated by ash are not consistently calibrated. This is to be expected: by setting df = infinity in ash (A0 and AH0), correction for small sample sizes no longer occurs, resulting in less conservative FDR estimates.

3.5.2 Pooled Fraction Spike-in Data

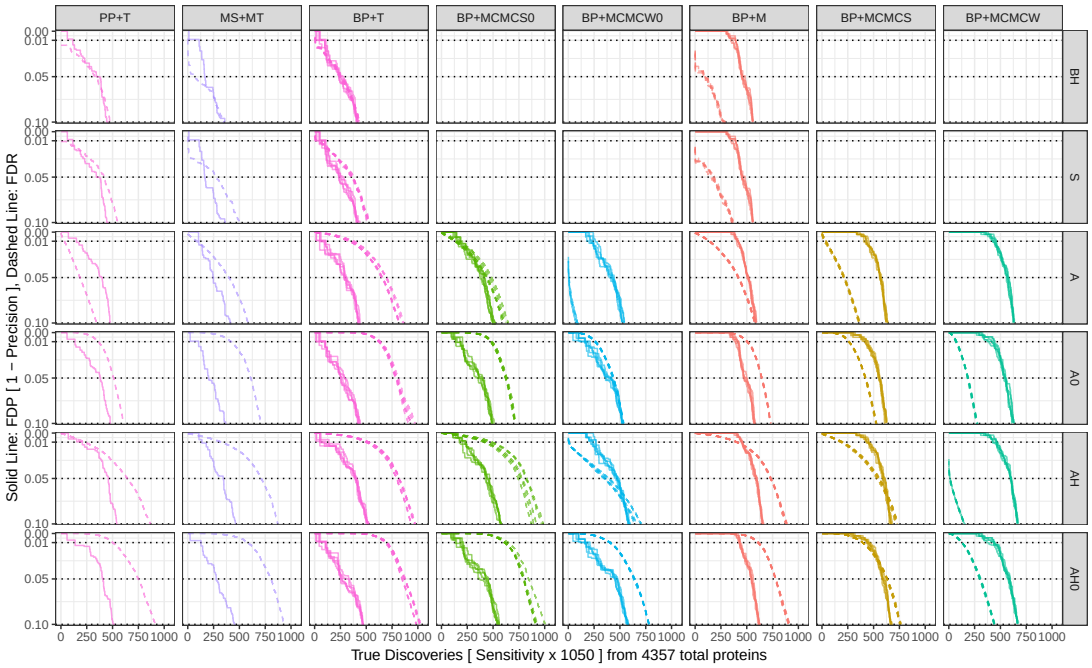

Figure 3.15: Precision-recall curves with FDR for pooled fraction spike-in data. Results of repeat runs of BayesProt with differing random seeds are shown as separate lines. FDP is shown as solid lines where FDR is shown with dashed lines.

For this more difficult data set, the recall of all methods is reduced compared to the single fraction data set, even after accounting for the smaller number of identified proteins. As above, the methods which incorporate the uncertainty of the protein quantifications show an increase in recall over the other methods. Similar trends to the single fraction data set are seen regarding the calibration of FDR: ProteinPilot and MSstatsTMT show calibrated FDR with Benjamini–Hochberg but not with Storey's q-value method and ash; the calibration of the FDR for the methods utilising uncertainty information is dependent on the configuration of ash used.

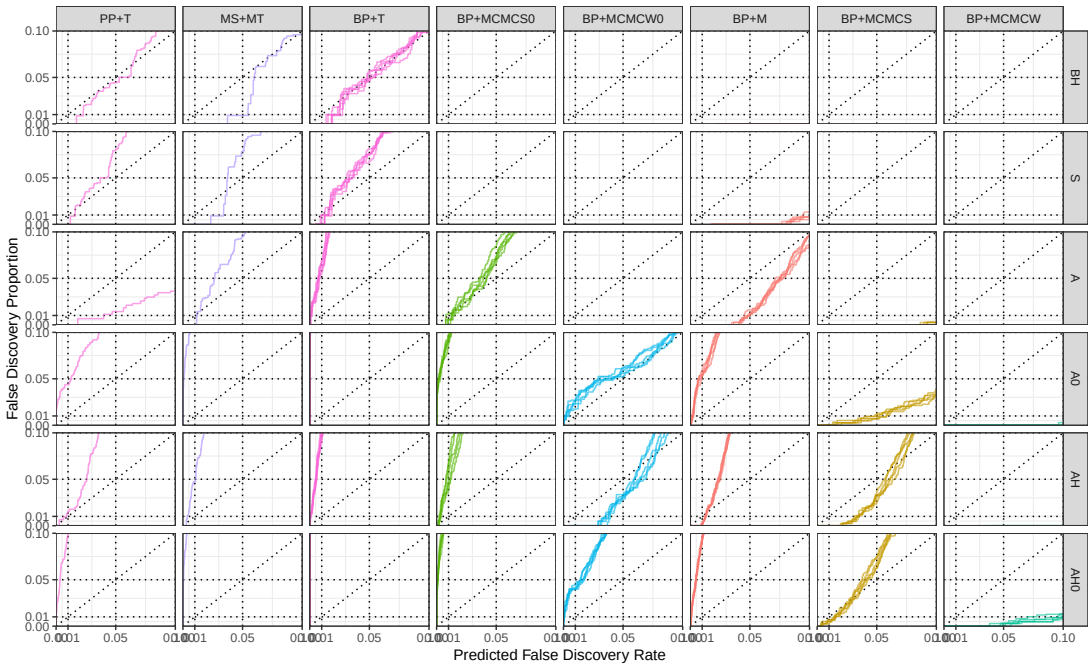

Figure 3.16: False Discovery Proportion vs False Discovery Rate curves for the pooled fraction spike-in data. Results of repeat runs of BayesProt with differing random seeds are shown as separate lines.

Table 3.12: Calibrated mean recalls of tested methods for the pooled fraction spike-in data

	Calibrated Recall		
Method	1% FDR	5% FDR	10% FDR
PP+T+BH		327	
PP+T+S			
PP+T+A	32	184	346
PP+T+A0			
PP+T+AH			
PP+T+AH0			
MS+MT+BH	1	72	347
MS+MT+S	2		
MS+MT+A	88		
MS+MT+A0			
MS+MT+AH			
MS+MT+AH0			
BP+T+BH	6.4 ± 0.548		
BP+T+S			
BP+T+A			
BP+T+A0			
BP+T+AH			
BP+T+AH0			
BP+MCMCS0+A			
BP+MCMCS0+A0			
BP+MCMCS0+AH			
BP+MCMCS0+AH0			
BP+MCMCW0+A		5.4 ± 1.14	84 ± 8.43
BP+MCMCW0+A0			
BP+MCMCW0+AH	5.4 ± 1.67	368.4 ± 21.7	
BP+MCMCW0+AH0			
BP+M+BH		93.2 ± 8.44	281 ± 7.91
BP+M+S		171.2 ± 14.5	$\mathbf{363.4 \pm 5.77}$
BP+M+A	137.8 ± 2.28	$\mathbf{419.6 \pm 2.79}$	$\mathbf{581.2 \pm 3.27}$
BP+M+A0			
BP+M+AH	$\mathbf{403.6 \pm 3.36}$		
BP+M+AH0			
BP+MCMCS+A	32.6 ± 7.7	214.2 ± 4.76	362.6 ± 5.13
BP+MCMCS+A0	$\mathbf{260.8 \pm 2.17}$	$\mathbf{421.4 \pm 3.05}$	$\mathbf{524.8 \pm 3.27}$
BP+MCMCS+AH	192.8 ± 7.85	$\mathbf{545.4 \pm 10.4}$	
BP+MCMCS+AH0	$\mathbf{393.8 \pm 0.447}$		
BP+MCMCW+A			
BP+MCMCW+A0	71.6 ± 3.65	184.6 ± 3.65	272.4 ± 5.08
BP+MCMCW+AH		11.8 ± 2.49	150.8 ± 2.05
BP+MCMCW+AH0	126.4 ± 2.3	303.2 ± 5.76	441.6 ± 3.51

Mean values at FDR cutoffs of 1%, 5% and 10% calculated where all replicated BayesProt runs exhibit calibrated FDR at that particular FDR cutoff.

3.5.3 Faulty Spike-in Data

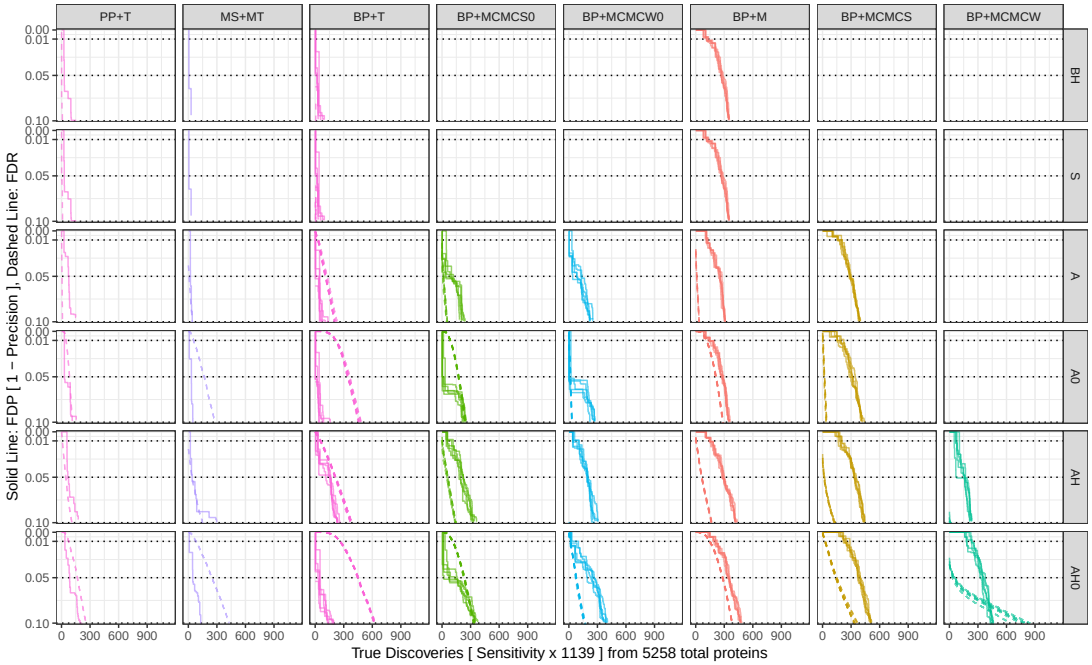

Figure 3.17: False Discovery Proportion and False Discovery Rate curves for the faulty Spike-in data versus True Discoveries. Results of repeat runs of BayesProt with differing random seeds are shown as separate lines. FDP is shown as solid lines where FDR is shown with dashed lines.

For the faulty data set, the recall of all methods is reduced further. In multiple instances, no proteins are found to be significant at 1%, 5% and 10% FDR. The methods which incorporate the uncertainty of the protein quantifications achieve the highest recall but, as above, the calibration of FDR is inconsistent. For the unequal variances Bayesian t-test (BP+MCMCW), in two instances (A and A0) ash assigns all proteins an FDR of 1.0.

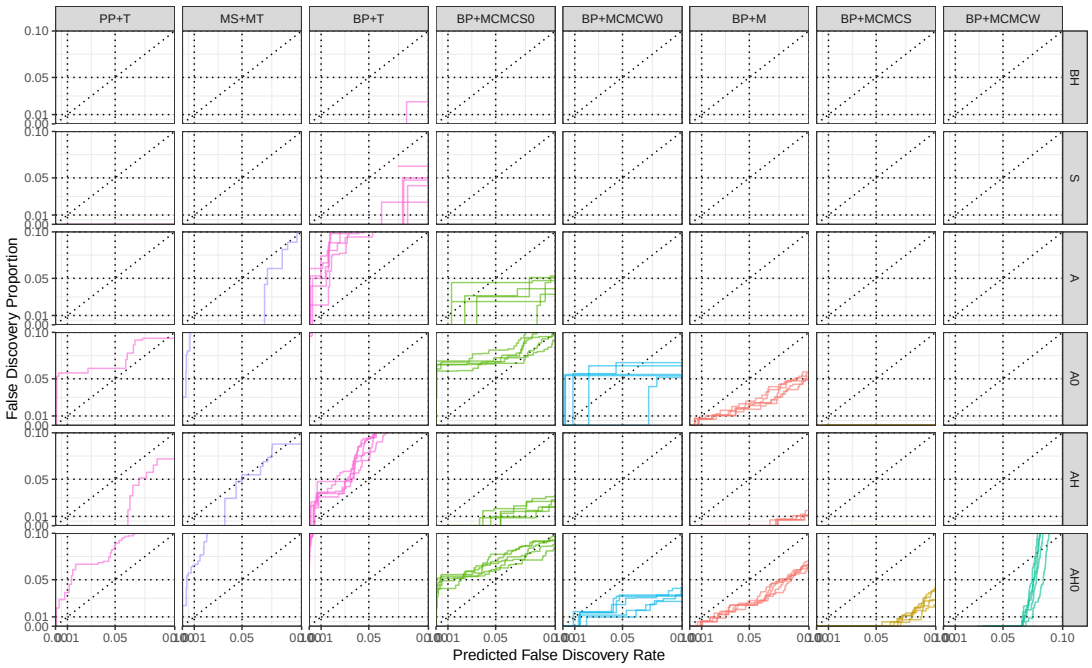

Figure 3.18: False Discovery Proportion vs False Discovery Rate curves for the faulty spike-in data. Results of repeat runs of BayesProt with differing random seeds are shown as separate lines.

Table 3.13: Calibrated mean recalls of tested methods for the faulty spike-in data.

	Calibrated Recall		
Method	1% FDR	5% FDR	10% FDR
PP+T+BH	1	1	9
PP+T+S	1	1	11
PP+T+A	1	6	10
PP+T+A0			115
PP+T+AH	9	45	109
PP+T+AH0			
MS+MT+BH			
MS+MT+S			
MS+MT+A		8	
MS+MT+A0			
MS+MT+AH		40	149
MS+MT+AH0			
BP+T+BH			
BP+T+S			5 ± 1.58
BP+T+A			
BP+T+A0			
BP+T+AH			
BP+T+AH0			
BP+MCMCS0+A		13.2 ± 5.4	51.8 ± 5.72
BP+MCMCS0+A0			
BP+MCMCS0+AH		56.2 ± 6.3	141.8 ± 4.6
BP+MCMCS0+AH0			$\mathbf{334.4 \pm 3.29}$
BP+MCMCW0+A			
BP+MCMCW0+A0			32.4 ± 1.52
BP+MCMCW0+AH			
BP+MCMCW0+AH0		76 ± 4.42	166.4 ± 5.94
BP+M+BH			
BP+M+S			
BP+M+A		10.4 ± 1.14	35.4 ± 0.548
BP+M+A0	$\mathbf{113.6 \pm 1.14}$	$\mathbf{211.8 \pm 1.92}$	286.6 ± 1.82
BP+M+AH	4 ± 0.707	63.6 ± 2.07	166.4 ± 3.13
BP+M+AH0	$\mathbf{147.2 \pm 1.92}$	$\mathbf{282.4 \pm 2.51}$	$\mathbf{382.8 \pm 1.48}$
BP+MCMCS+A			
BP+MCMCS+A0	3.8 ± 1.92	19.4 ± 2.88	41 ± 3.16
BP+MCMCS+AH		17.4 ± 2.3	126.6 ± 4.39
BP+MCMCS+AH0	$\mathbf{26.8 \pm 2.86}$	$\mathbf{143.6 \pm 5.41}$	$\mathbf{348.2 \pm 16}$
BP+MCMCW+A			
BP+MCMCW+A0			
BP+MCMCW+AH			8.6 ± 6.8
BP+MCMCW+AH0		47 ± 9.82	

Mean values at FDR cutoffs of 1%, 5% and 10% calculated where all replicated BayesProt runs exhibit calibrated FDR at that particular FDR cutoff.

3.5.4 Discussion of FDR Results

Benjamini–Hochberg and Storey's q-value

Comparing the false discovery rate estimates between the Benjamini–Hochberg (BH) and Storey's q-value (S) on the applicable quantification methods which output p-values (PP+T, BP+T, BP+M), it is clear that the Benjamini–Hochberg procedure is more conservative than Storey's qvalue. In the cases of the standard t-test on both the ProteinPilot quantifications (PP+T) and median BayesProt quantifications (BP+T), application of Storey's q-value method leads to a loss of FDR control: the estimated FDR is greater than the ground truth FDP for parts of the range.

For the metafor t-test, the move to using Storey's q-value is not detrimental as above. The q-value method does lead to less conservative FDR estimates, though without the loss of FDR control: the FDR estimates are consistently higher than the ground-truth FDP.

For the spike-in data considered here, the FDR reported by Storey's method for the p-values generated by Student's t-tests on both the median BayesProt protein quantifications, ProteinPilot protein quantifications and MSstatsTMT's moderated t-test are generally anti-conservative; that is, the reported FDR is less than the actual observed FDP.

Conversely, the FDR reported by Storey's method and Benjamini–Hochberg applied to the p-values generated by the metafor t-test are extremely conservative, that is, the reported FDR is greater than the FDP. While this is certainly preferable to being anti-conservative, in a clinical study where the true FDP is unobservable and standard practice is to consider, for example, the subset of proteins with FDR < 0.01 or FDR < 0.05, this would result in vastly reduced recall of differentially expressed proteins. The combination of the metafor t-test and these multiple testing correction methods are overly cautious.

The ash R package

The effect of the configuration of the ash package can be compared along two axes: the choice of mixture distribution, and the degrees of freedom parameter.

Using a half-uniform mixture distribution over a uniform mixture distribution, in effect allowing for non-symmetric distributions of effect estimates across the population of proteins, leads to less conservative false discovery rate estimates for all of the tested quantification methods. In some cases, this results in a loss of FDR control, as shown by comparing the A and AH rows for the metafor t-test (BP+M) method.

The effect of setting the *df* parameter in ash is now apparent: fixing *df* to be infinity and implying the normality of the quantification estimates results in less conservative estimates of FDR (as can be seen by comparing the curves for A with those for A0, and AH with AH0). Accounting for heavier-tailed quantification estimates leads to more conservative FDR.

For the simple t-test on median BayesProt quantification estimates, ash fails to control the FDR in all cases. However, this is unlikely to be the fault of the ash method; these t-tests are calculated only on the median protein quantification estimates, and hence the uncertainty in the effect estimate (from the `stderr` output of the `t.test` function in R) which is subsequently used by ash is based on incomplete information. This strengthens the argument for the correct propagation of uncertainty through a quantification pipeline.

The Bayesian t-tests which do not account for uncertainty (MCMCS0 and MCMCW0) are of less interest than their counterparts which incorporate the uncertainty (MCMCS and MCMCW); as noted above in Section 3.4, these methods exhibit a reduced recall over the more advanced methods. It is still useful to note that the effects of the configuration of the ash method are still apparent on these methods: the normality assumption (A0 and AH0) and non-symmetric mixture distribution (AH and AH0) make FDR estimates less conservative.

For the metafor t-test, ash only controls the FDR in one case, with a uniform mixture distribution and with the *df* parameter estimated per-protein. With a half-uniform mixture distribution, FDR control is lost.

There are now clear differences in the results from the Bayesian t-test with equal and unequal variances: for any of the given ash configurations, the choice of unequal variances over equal variances results in more conservative estimates of FDR. In combination with the reduced recall, this results in significantly lower calibrated recall at higher precisions (see Table 3.11).

While the Bayesian t-test with equal variances achieves the best recall out of all the methods tested, the reported FDR is rather conservative: the ground-truth FDP is much less than the reported FDR. Hence, when the calibrated FDR values are considered, the method does not perform as well.

For the purposes of exploratory analysis, where FDR is often employed for filtering of results over the already much stricter family-wise error rate (FWER) control provided by the likes of Bonferroni correction, overly strict FDR control is not necessarily desirable, as this reduces the number of candidates for further analysis.

3.6 Conclusions

BayesProt represents a current state-of-the-art tool for protein quantification, using the maximal amount of information in a hierarchical model to generate protein quantification estimates, including a measure of the uncertainty in the estimates. It has been demonstrated that the propagation of this uncertainty into the differential expression analysis is crucial for accurate inference.

Furthermore, it has been shown that expanding the empirical Bayes procedure to inform the prior distributions of all proteins, rather than the subset with fewer peptides or features, provides the best performance of the model in terms of the number of proteins correctly identified as being differentially expressed at relevant precision thresholds.

3.6.1 Future Work

Full Hierarchical Model

The empirical Bayes procedure applied in Section 3.4 can be conceptualised as an approximation to an ideal, fully hierarchical model where all proteins are analysed at once. Previous work has achieved this[38], with the added ability to estimate the proportion of proteins that are not differentially expressed, but with a less complex model. Analysis of all proteins at once would likely lead to better estimation of the false discovery rate. However, fitting such a model with MCMC would be much more computationally expensive than the current approach, where proteins are analysed separately.

3.6.2 Integration of Protein Identification and Quantification

The incorrect identification of peptide spectra is one of the possible causes for highly variable peptides at the quantification stage of an analysis pipeline. There have been some efforts in recent years[115][42] to combine the normally separate protein identification and quantification stages; a fully Bayesian approach to this, combining ambiguous peptide candidates from the identification stage with quantification data, would perhaps better infer the probability of correct identification and lead to better quantification estimates.

3.6.3 Motivations for Following Chapters

More broadly, there are a number of ways in which the results in this chapter can be improved upon, which will form the focus of the following chapters.

Computational Cost

Depending on the size of the data set and number of parameters being fitted, on a powerful multi-core workstation with excess RAM, BayesProt can still take a number of hours to run multiple MCMC chains to generate estimates for protein quantification. For example, for the analysis of the spike-in data set analysed in this chapter, one replicate of BayesProt running in its default configuration required approximately one hour to complete.

Of the methods for differential expression evaluated above, the better performing ones require yet more MCMC sampling, meaning more computational effort is needed to perform differential expression testing.

Chapter 4 will explore one possible mechanism by which the MCMC sampling could be completed with faster computation by marginalising so-called nuisance parameters to increase the efficiency of MCMC sampling.

Chapter 5 takes this a step further: the use of totally conjugate models eliminates the need for MCMC sampling for differential expression entirely.

False Discovery Rate Estimation

Beyond the requirement of allowing for the additional uncertainty information, there is no clear "best" choice of algorithm for differential expression and FDR estimation; the tested methods force the user to make trade-offs in terms of either sacrificing recall at lower or higher precisions, or guarantees of controlled false discovery rate.

For example, the Bayesian t-test method that generally provides the best ordering of proteins, when combined with the ash FDR estimation, tends to give much lower recall at any given FDR cutoff since the estimated FDRs are overly conservative.

Chapter 5 seeks to improve on these methods by employing a model comparison testing method via the calculation of Bayes factors. The analytic calculation of Bayes factors requires that models be analytically tractable and ensuring this property has the additional side effect of allowing for rapid estimation of the posterior distribution without the need for computationally expensive MCMC, reducing the time required to complete differential expression testing.

Analysis of Shared Peptides

Another natural extension of the model would be to make provisions to allow for the analysis of shared peptides; the model described above is suited only for the analysis of unique peptides. Including information from shared peptides in quantitative analysis has previously been demonstrated to improve the accuracy of protein quantification estimates[48][116].

The MCMCglmm R package provides a simple interface to a Gibbs sampler, allowing users to quickly generate results without the need for the expert knowledge required to develop their own sampler. However, MCMCglmm has been designed with a limited class of model in mind; the non-linear modelling required for shared peptide analysis is out of scope for the package.

In Chapter 6 a variant of the BayesProt model is developed with the aim to allow for the analysis of shared peptides, using the Stan programming language[81] to specify and fit a more complex model than MCMCglmm would normally allow.

Chapter 4

Increased Efficiency of Poisson Model Using Conjugate Priors

4.1 Gamma-based Model

BayesProt represents a current state-of-the-art model for protein quantification. However, its log-normal priors with a Poisson likelihood do not form a conjugate system; that is, all integrals are intractable, meaning that every variable must be sampled in order to build up an approximation of the full posterior, including so-called "nuisance" variables which we have little or no interest in, but which are necessary for the correctness of the model. More efficient sampling behaviour can be obtained if models are constructed in such a way that the conjugate relationship between probability distributions (namely, the Poisson and gamma distributions) can be exploited, making some of the integrals analytic, and thereby reducing the size of the sample space.

Furthermore, the ability to analyse shared peptides, peptides which could have arisen from more than one protein, is desirable. In order for this to be correctly modelled, the underlying "true" abundance of each protein which has a shared peptide must be considered, rather than simple ratios of ion intensities. The result is a non-linear model, which the Poisson mixed-effects modelling in MCMCglmm[104] used in BayesProt cannot accommodate.

The combination of these two factors motivated the development of a new model. As discussed above in Section 2.1.6, ion counts can be demonstrated to be Poisson distributed so the model that is described in this chapter has been developed around this simple assumption.

4.2 Background Theory

The definitions of the probability distributions used below can be found in Appendix A.

Suppose for a given LC-MS feature in sample k with abundance a_k, y_k ions are observed:

$$y_k \sim \text{Poisson}(a_k) \tag{4.1}$$

Then by assigning a gamma prior to the abundance a_k the fact that the gamma distribution is a conjugate prior to the Poisson distribution can be exploited.

$$a_k \sim \text{Gamma}(\alpha, \beta) \tag{4.2}$$

Then the posterior of a_k is also a gamma distribution:

$$\rho(a_k|y_k) = \text{Gamma}(\alpha + y_k, \beta + 1) \tag{4.3}$$

More generally, for N observations $\mathbf{y_k} = (y_{1,k}, \ldots, y_{N,k})$ and this generalises to:

$$\rho(a_k|\mathbf{y}_k) = \text{Gamma}\left(\alpha + \sum_{i=1}^{N} y_{i,k}, \beta + N\right) \tag{4.4}$$

Then the posterior predictive distribution, the probability distribution which describes the distribution of a subsequent $(N+1)$th observation, ${y_k}'$, of the same feature given that there have already been N observations, $\mathbf{y}_k$, takes the form of a negative binomial distribution:

$$\rho({y_k}'|\mathbf{y_k}) = \text{NegBin}\left(\alpha + \sum_{i=0}^{N} y_{i,k}, \frac{1}{1 + (\beta + N)}\right) \tag{4.5}$$

These results will allow for sampling to be performed on a smaller parameter space by integrating out the variables we have no interest in (see Section 4.3.1).

4.3 Methods

Suppose then that there are two observations of a peptide, one from each of two samples, $k = 1$ and $k = 2$:

$$y_{k=1} \sim \text{Poisson}(a_1) \tag{4.6}$$

$$y_{k=2} \sim \text{Poisson}(a_2) \tag{4.7}$$

In quantitative proteomics studies where, due to differences in ionisation of peptides, only relative quantification is possible, only the ratio between a_1 and a_2 is of interest, rather than their individual values. Let this ratio equal C:

$$C = \frac{a_2}{a_1} \tag{4.8}$$

Instead of placing a prior distribution directly on a_2, an arbitrary prior distribution is placed on the *change* between a_1 and a_2, C.

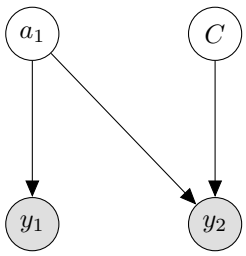

Figure 4.1: Bayesian network for the toy example considered here, showing the relationship between the observed counts y_1 and y_2, the abundance of the protein in the control group, a_1, and the change in abundance between the control and treatment groups, C.

In reality, C would also incorporate changes arising from systematic differences between peptides and charge states, random variation due to digestion among others, and as such its prior may be a combination of a number of different priors accounting for these factors.

It then follows that:

$$y_1 \sim \text{Poisson}(a_1) \tag{4.9}$$

$$y_2 \sim \text{Poisson}(a_1 \cdot C) \tag{4.10}$$

This model can be fitted trivially with a number of algorithms, including MCMC.

4.3.1 Collapsed Sampler

Suppose that a count y_1 is observed first, before a count y_2 is observed. After observing y_1, the posterior on a_1 is:

$$\rho(a_1|y_1) \sim \text{Gamma}(\alpha + y_1, \beta + 1) \tag{4.11}$$

Then given that at any point the value of C (which is just some scalar value we have sampled) is known for this MCMC iteration, the posterior on $a_1 \cdot C$ conditional on y_1 is, via the scaling property of the gamma distribution:

$$\rho(a_1 \cdot C|y_1, C) \sim \text{Gamma}\left(\alpha + y_1, \frac{\beta + 1}{C}\right) \tag{4.12}$$

The posterior predictive distribution of y_1 given no prior observations is:

$$\rho(y_1) = \text{NegBin}\left(\alpha, \frac{1}{1 + \beta}\right) \tag{4.13}$$

The posterior predictive distribution of y_2 given that y_1 has already been observed is:

$$\rho(y_2|y_1, C) = \text{NegBin}\left(\alpha + y_1, \frac{1}{1 + (\frac{\beta+1}{C})}\right) \tag{4.14}$$

The probability density of interest is the posterior of C conditional on y_1 and y_2, $\rho(C|y_1, y_2)$. Then via Bayes theorem we have:

$$\rho(C|y_1, y_2) \propto \rho(C) \cdot \rho(y_1|C) \cdot \rho(y_2|y_1, C) \tag{4.15}$$

$$= \rho(C) \cdot \rho(y_1) \cdot \rho(y_2|y_1, C) \tag{4.16}$$

The terms $\rho(y_1)$ and $\rho(y_2|C, y_1)$ are simply the Negative Binomial distributions 4.13 and 4.14 above, and $\rho(C)$ is an arbitrary prior on C.

Here, a_1 is integrated out. This is not a problem since in order to estimate relative quantification of a peptide between two samples the value of a_1 is not of interest, only the value of C. This reduces the number of dimensions of the parameter space, allowing for more efficient sampling with reduced autocorrelation between successive samples; with the same number of MCMC iterations, this collapsed version of the sampler produces more accurate estimates of C than the full naïve sampler.

A marginal posterior for a_1 can still be obtained. For each MCMC iteration a sample from the posterior of C is obtained and it can be demonstrated that the posterior of a_1 conditional on C is:

$$\rho(a_1|y_1, y_2, C) \sim \text{Gamma}(\alpha + y_1 + y_2, \beta + 1 + C) \tag{4.17}$$

that is, for each MCMC iteration and sample from C, there is a corresponding gamma distribution describing the posterior of a_1 for that iteration. Hence the full marginal posterior of a_1 is simply a mixture of these gamma distributions. This has the additional benefit of allowing a_1 to be estimated to a greater degree of accuracy for the same number of samples than the naïve model, since any statistics calculated over a population of distributions are more accurate than those same statistics calculated across a population of individual samples.

4.3.2 Test Framework

To demonstrate the increase in sampling efficiency, i.e. the increased number of effective independent samples generated per second, that is possible when some parameters are marginalised the Stan programming language[81] is used to fit equivalent models to simulated data. Stan provides a fast and efficient implementation of a Hamiltonian Monte Carlo sampler, representing a current state-of-the-art MCMC sampler.

The first model is parameterised in the naïve fashion, with a Poisson likelihood, a gamma prior on the base Poisson rate, λ, and a log-normal prior on the change between groups, C:

$$y \sim \text{Poisson}(\lambda \cdot \mathbf{X} \cdot C) \tag{4.18}$$

$$\lambda \sim \text{Gamma}(\alpha, \beta) \tag{4.19}$$

$$C \sim \log\text{-Normal}(0, 100) \tag{4.20}$$

where X is a design matrix indicating which group each element of y belongs to.

The marginalised model takes the form:

$$y_i \sim \text{Negative Binomial}\left(\alpha + \sum_{j=1}^{i-1} y_j, \frac{\beta + \sum_{j=1}^{i-1} (\mathbf{X} \cdot C)_j}{(\mathbf{X} \cdot C)_i}\right) \tag{4.21}$$

$$C \sim \log\text{-Normal}(0, 100) \tag{4.22}$$

such that the likelihood for each y_i is dependent on the sum of all previous observations $y_{1:i-1}$ and the sum of inferred changes for each of those observations.

Results from six sets of simulated data are presented: the first consists of a 5 vs 5 comparison, that is, with one change parameter being estimated; the second and third consist of a 15 vs 15 and a 60 vs 60 comparison i.e. with a single change parameter but with a larger number of observed data points. The fourth consists of four groups of five points, the fifth with six groups of five points, and the sixth with 24 groups of five points; that is, with 3, 5 and 23 change parameters being estimated relative to the first group. The six simulated data sets were chosen to investigate how the marginalised model's statistical speedup scales, given data size and dimensionality of the parameter space.

Three variants of the marginalised model are tested which differ only in how the cumulative sum operation across change parameters was implemented: firstly, the cumulative sum is performed with a matrix multiplication operation; secondly with sparse matrix multiplication; and finally using a simple for-loop within the Stan program. It should be noted that the cumulative sum of y can simply be calculated before sampling and included as additional observed data, so this operation has a negligible computational cost.

For each data set, the models were run using the Stan programming language with the CmdStan.jl[117] interface for the Julia[118] programming language. Each model was ran for 20,000 MCMC iterations on each of 4 chains, the first 10,000 iterations being used as warm-up iterations and hence discarded before calculating results. This was repeated 100 times for each model and data set with Stan initiated with a different

random seed for each repetition. Timings were recorded using Julia's built-in timing macro `@time`, and effective sample size was calculated as the sum of effective sample sizes for each change parameter across the four MCMC chains. The effective sample size algorithm was the same[71] as that used in Stan's `stansummary` program but rewritten in the Julia language.

4.4 Results

Plots of:

- effective sample size (ESS)
- runtime
- effective sample size per second (ESS/s), a measure a sampling efficiency
- statistical speedup relative to the naïve model

are presented for each model and for each simulated data set described above. The "statistical speedup" is calculated as the ratio of the sampling efficiency of a model versus the sampling efficiency of the naïve model.

The models are denoted in each plot as follows:

- Naïve — naïve Poisson likelihood model
- M-MM — marginalised model with matrix multiplication for cumulative sum
- M-SM — marginalised model with sparse matrix multiplication for cumulative sum
- M-Loop — marginalised model with simple loop for cumulative sum.

Figure 4.2 presents violin plots of the mean ESS across the estimated change parameters for each model and data set. With increased numbers of data points, the ESS for all models remains stable. However, as the number of parameters to be estimated increases, the mean ESS decreases; sampling from a higher dimensional posterior is less efficient. The three versions of the marginalised model show increased ESS over the naïve model and are comparable in their efficiency.

Figure 4.3 presents violin plots of the runtime for each model and data set. The marginalised model with the simpler for-loop demonstrates improved runtime over the other implementations of the model, both for large numbers of data points and for

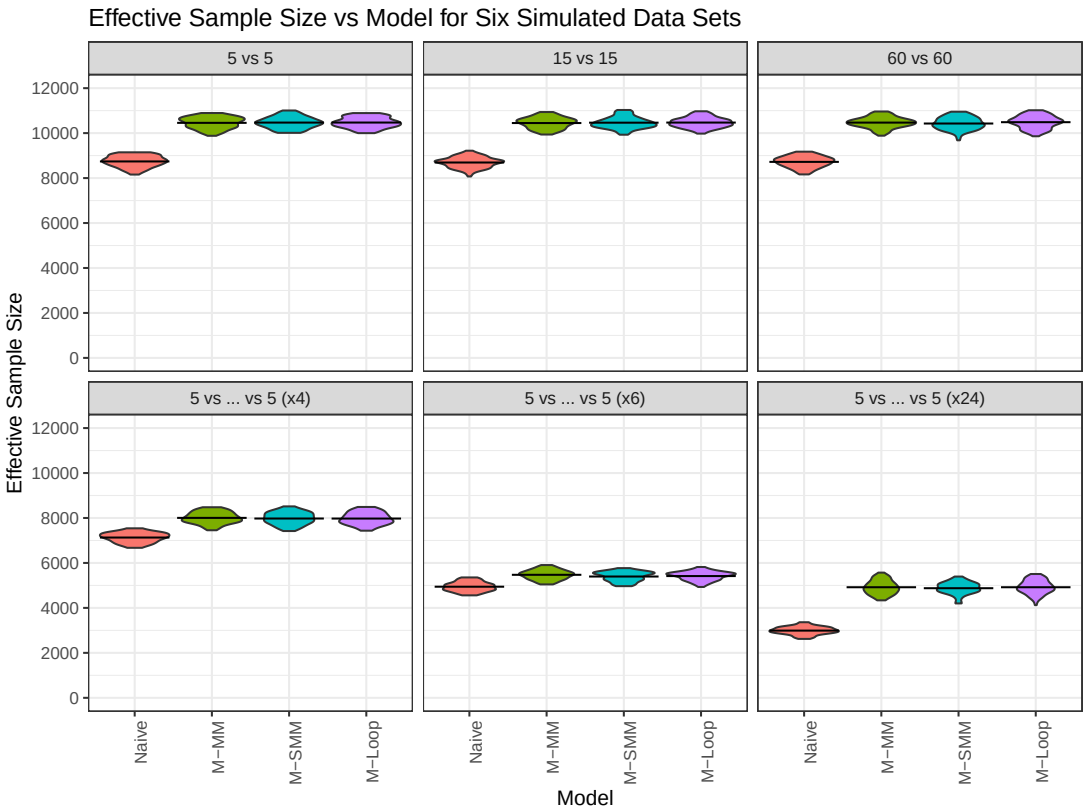

Figure 4.2: Effective sample size vs model for six simulated data sets — Violins showing density of mean ESS across the estimated parameters for 100 repetitions of each model with differing random seeds for each of the six data sets described above.

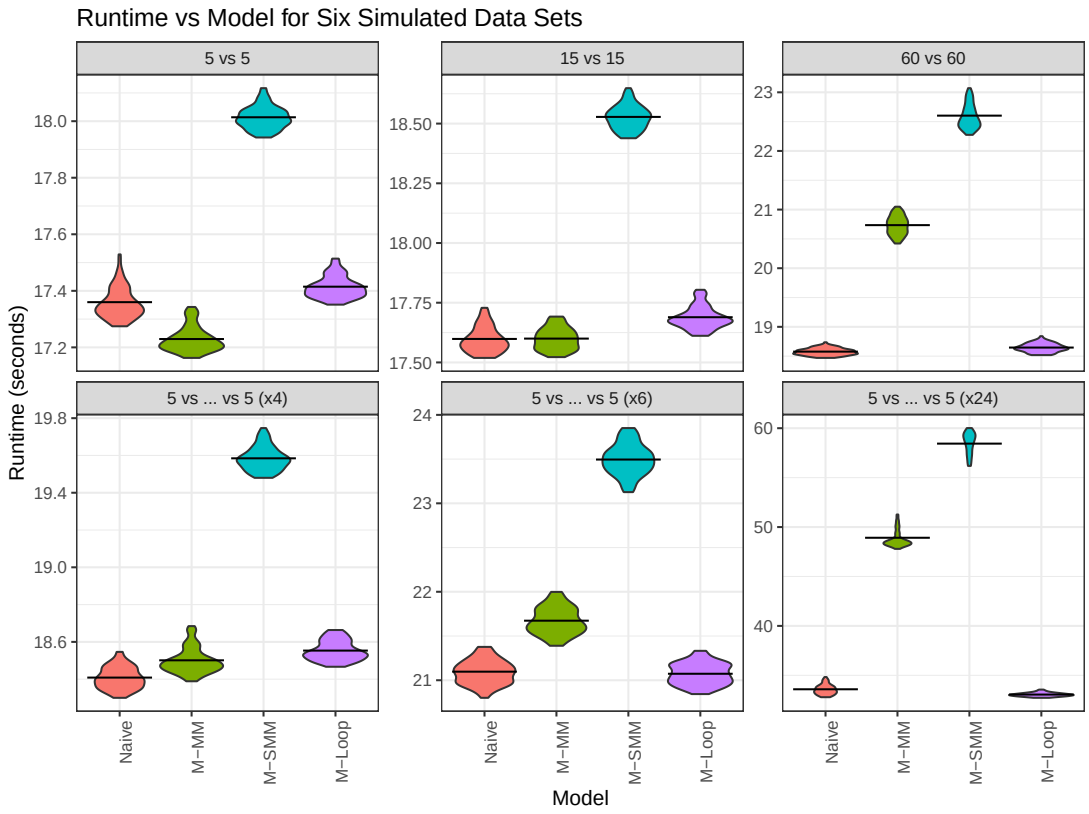

Figure 4.3: Runtime vs model for six simulated data sets — Violins showing density of runtime for 100 repetitions of each model with differing random seeds for each of the six data sets described above. Note the differing y-axis scales for each data set.

larger numbers of estimated parameters. The matrix multiplication and sparse matrix multiplication versions of the marginalised model both fail to scale with numbers of data points and with numbers of estimated parameters.

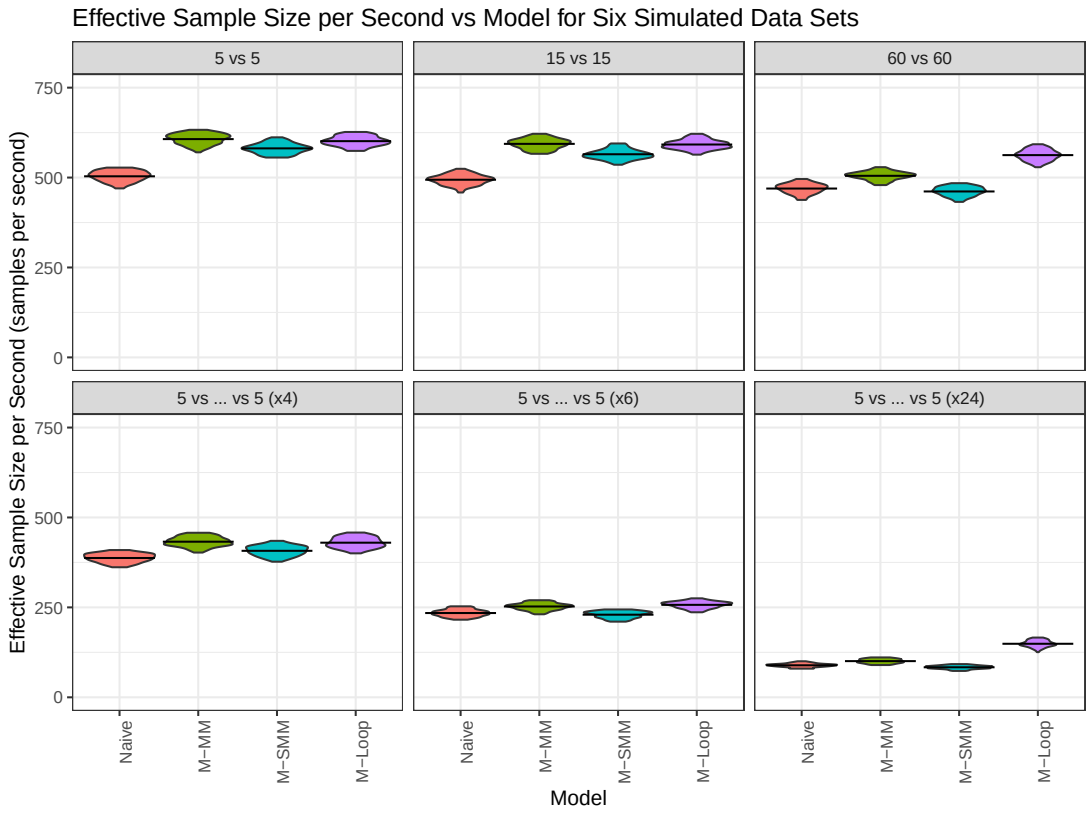

Figure 4.4: Effective sample size per second vs model for six simulated data sets — Violins showing density of mean ESS per second across the estimated parameters for 100 repetitions of each model with differing random seeds for each of the six data sets described above.

Figure 4.4 presents violin plots of mean ESS per second across the estimated parameters for each model and data set. For larger data sets and with more parameters to be estimated, the marginalised model with the simple for-loop for cumulative sum demonstrates a greater sampler efficiency than the other implementations of the model.

Figure 4.4 presents violin plots of mean ESS per second across the estimated parameters for each model and data set relative to the median of the mean ESS per second across the estimated parameters for the naïve model. The marginalised model that utilises a simple for-loop demonstrates a statistical speedup over the naïve model, which becomes evident for larger numbers of data points and for larger numbers of

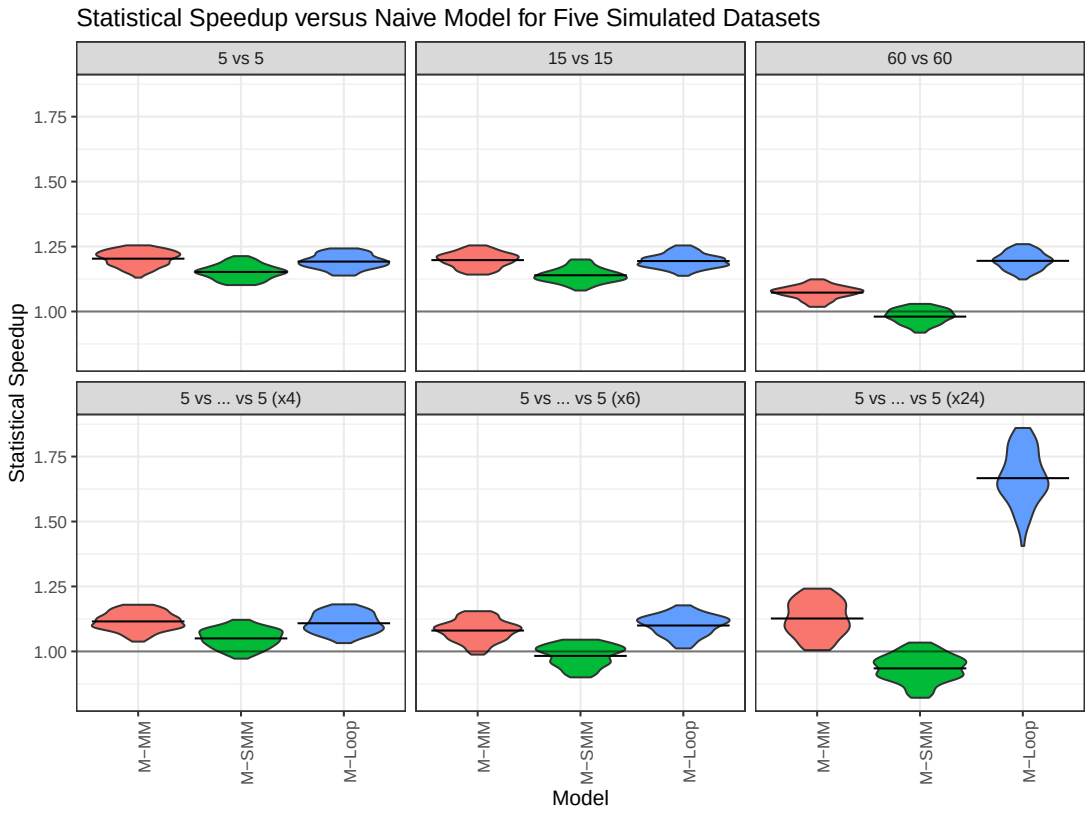

Figure 4.5: Statistical speedup vs model for six simulated data sets — Violins showing density of statistical speedup relative to median of the mean ESS per second across the estimated parameters of naïve model for 100 repetitions of the marginalised models with differing random seeds for each of the six data sets described above.

estimated parameters.

4.5 Discussion

It is immediately obvious that for smaller sets of data with very few parameters, for the same number of MCMC iterations, the collapsed samplers obtain higher effective sample sizes (and thereby smaller estimation error) for the parameters of interest and therefore the unsampled parameter a than the naïve model with a Poisson likelihood.

However, as the size of the data set or number of model parameters increases, the gains in sampling efficiency which come as a result of the marginalisation of parameters do not scale for all implementations of the model: for the implementations utilising matrix multiplication to perform the cumulative sum operation, the statistical speedup decreases as the number of data points increases (as can be seen by comparing results for data set 2 and 3 with data set 1), and also decreases when the dimension of the parameter space is increased (as can be seen by comparing results of data set 4, 5 and 6 versus data set 1).

The statistical speedup for the model using a for-loop to perform the cumulative sum operation is more evident for the data sets with larger numbers of datapoints (data set 3) and with larger numbers of parameters (data set 6). This speedup is maintained with increasing amounts of data (data sets 2 and 3) but begins actually increases as the size of the parameter space increases (data sets 4 and 5); for the naïve model, data sets 4, 5 and 6 require sampling from higher dimensional parameter spaces of four, six and 24 dimensions respectively. Marginalising over one of these dimensions represents a smaller reduction in dimensionality (25%, 16% and 4.17%) than in data sets 1 through 3, where the relative reduction in dimensionality is higher (50%). The additional computation required to marginalise over this one dimension is outweighed by the increase in sampling efficiency gained by the marginalisation.

4.6 Conclusions

Hamiltonian Monte Carlo via Stan generates samples from a posterior distribution with low auto-correlation, resulting in high effective samples sizes for parameters. In the cases where models can be reparameterised to exploit conjugate distributions, marginalisation of parameters can grant an increase in the statistical efficiency of a given sampler. However, for complex models it may be necessary to perform additional computation within the sampler in order to exploit this conjugacy.

In this chapter it has been demonstrated for a simple class of models relevant to quantitative proteomics studies that there are some efficiency gains to be made. By

marginalising out one or more of the parameters in the model, the number of effective samples generated by a sampler each second is increased over a naïve implementation which fits data to the full model with all parameters represented. Hence, estimation of a model to the desired degree of accuracy can be achieved with fewer MCMC iterations.

Additionally, it has been shown that while this increase in sampler efficiency is evident for small models with small amounts of data, the additional computation required to utilise the marginalised model renders the marginalised model less efficient than its naïve counterpart when that computation involves multiplication of large matrices. Using a simple loop to perform this additional computation allows the marginalised model to scale in higher dimensions.

We note that similar marginalisations might be performed on other model classes (e.g. conjugate normal-inverse-gamma priors on the mean and variance of a normal distribution) that also require cumulative sums of other sampled parameters, and we would therefore expect similar behaviour of marginalised models.

These conclusions motivate the approach taken in the following chapter, Chapter 5: by constructing a model for differential expression testing consisting entirely of conjugate distributions, Bayesian estimation of the model can be achieved without the need for computationally expensive sampling algorithms such as MCMC.

Chapter 5

Calibration of Quantitative False Discovery Rate Through Model Comparison Testing

5.1 Introduction

When selecting sets of candidate proteins in a quantitative proteomics study, it is essential that the false discovery rate (FDR) is controlled to an acceptable level. In Chapter 3, analysis of spike-in studies showed that current methods often fail to provide well-controlled estimates of the false discovery rate, instead often either underestimating the proportion of false positives in the significant set, or overestimating the proportion, leading to reduced recall when results are thresholded by FDR.

This chapter describes a fully Bayesian methodology for differential expression testing, which has the potential to produce more accurate estimates of FDR by using Bayes factors to generate posterior error probabilities for each protein. The ability of this method to give more accurate FDR estimates on a number of spike-in and simulated data sets is demonstrated.

Section 5.2 describes a pair of models for differential expression testing which are to be compared, before then explaining how Bayesian inference on these models can be performed analytically without the need for numerical integration such as MCMC.

Section 5.3 makes justifications for the prior parameters adopted for the Bayesian model comparison, explaining their significance.

Chapter 3 highlighted the importance of making use of the additional information of protein quantification uncertainty. In Section 5.4, an approximate method for

incorporating this uncertainty into the analytic model is described.

Section 5.5 describes the data and methods used to analyse the Bayesian model comparison and similar methods for their ability to both correctly identify differentially expressed proteins in a data set and provide calibrated estimates of FDR.

Section 5.6 presents results from the tested methods, similar to that shown in Chapter 3.

Section 5.7 presents results demonstrating the speedup of the Bayesian model comparison over QPROT[43], a software tool which utilises a similar linear model fitted with MCMC to perform differential expression testing as an example of the reduction in computation time afforded by the analytic approach taken in this chapter.

Section 5.8 makes some conclusions based on the results in Sections 5.6 and 5.7.

Finally, Section 5.9 identifies some potential goals for future research.

5.2 Models

In Chapter 3 a number of methods for differential expression testing were considered, some of which used MCMC to fit a linear model forming a Bayesian analogue to Student's and Welch's t-tests. This chapter considers a reformulation of that model which is amenable to conjugate analysis, with the objective of performing Bayesian model comparison testing to infer the probability that each protein is differentially expressed. In order to make comparisons between models, naturally multiple models must be considered.

5.2.1 Model Comparison

We consider the simplest applicable full model M_F, where we observe protein-level log-intensities y and z from two groups of samples with some log-fold-change d between the mean intensities of the two groups.

$$P(y|M_F) = \mathrm{N}(\mu, \sigma^2) \tag{5.1}$$
$$P(z|M_F) = \mathrm{N}(\mu + d, \sigma^2) \tag{5.2}$$
$$P(\mu|M_F) = \mathrm{N}(\mu_\mu, \sigma_\mu^2) \tag{5.3}$$
$$P(d|M_F) = \mathrm{N}(\mu_d, \sigma_d^2) \tag{5.4}$$
$$P(\sigma^2|M_F) = \text{Scale-Inv-}\chi^2(\nu, \tau^2) \tag{5.5}$$

In tandem with this, we consider a null model M_N, which is exactly the same but with the exception that the log-fold-change is fixed at 0, that is, there is no observable change.

$$P(y|M_N) = \mathrm{N}(\mu, \sigma^2) \tag{5.6}$$
$$P(z|M_N) = \mathrm{N}(\mu, \sigma^2) \tag{5.7}$$
$$P(\mu|M_N) = \mathrm{N}(\mu_\mu, \sigma_\mu^2) \tag{5.8}$$
$$P(\sigma^2|M_N) = \text{Scale-Inv-}\chi^2(\nu, \tau^2) \tag{5.9}$$

The null model is a nested, special case of the full model where $d = 0$.

The posterior probability of a model M_x given the observed data $D = (y, z)$ can be written by Bayes rule:

$$P(M_x|D) = \frac{P(D|M_x) \cdot P(M_x)}{P(D)} \tag{5.10}$$

Ratioing the probability of the null model versus the full model gives the expression:

$$\frac{P(M_N|D)}{P(M_F|D)} = \frac{P(D|M_N)}{P(D|M_F)} \cdot \frac{P(M_N)}{P(M_F)} \tag{5.11}$$

The term $\frac{P(D|M_N)}{P(D|M_F)}$ is known as the marginal likelihood ratio or the Bayes Factor B_{NF}. The term $\frac{P(M_N)}{P(M_F)}$ represents the relative prior probabilities of each model. Assuming that only these two models are possible, (that is, either the protein is differentially expressed or it is not and assuming that the rest of the model is correct, i.e normally distributed residuals) by the law of total probability, $P(M_N|D) = 1 - P(M_F|D)$:

$$\frac{1 - P(M_F|D)}{P(M_F|D)} = \frac{P(D|M_N)}{P(D|M_F)} \cdot \frac{P(M_N)}{P(M_F)} \tag{5.12}$$
$$\implies P(M_F|D) = \frac{1}{\left(1 + \frac{P(D|M_N)}{P(D|M_F)} \cdot \frac{P(M_N)}{P(M_F)}\right)} \tag{5.13}$$

From these values, a probability that the full model M_F is the "correct" one (i.e. a probability that it best explains the observed data), $P(M_F|D)$, can be calculated conditional on the data, $D = (y, z)$, the log-protein-intensities. This probability is

dependent on terms that can either be calculated or have prior values fixed for. For simplicity, the assumption that both models are equally likely is made and hence the term $\frac{P(M_N)}{P(M_F)}$ is set to 1.

Finally, to create a value with a similar direction to a p-value (with smaller values indicating a higher probability of differential expression), we calculate the "posterior error probability" (PEP) for each protein. The PEP has seen use for identification error rates in proteomics[119], but here is used for the error rate of quantification. The PEP can be calculated for each protein as:

$$PEP = 1 - P(M_F|D) \tag{5.14}$$

$$= 1 - \frac{1}{\left(1 + \frac{P(D|M_N)}{P(D|M_F)} \cdot \frac{P(M_N)}{P(M_F)}\right)} \tag{5.15}$$

Generating this PEP is reliant on being able to calculate or estimate the marginal likelihood ratio of the two models given the observed data.

For models fitted with MCMC sampling, the Bayes factor or marginal likelihood ratio can be estimated from the posterior samples through a number of methods, including the harmonic mean estimator of Newton and Raftery[86], the Savage–Dickey density ratio[91] and bridge sampling[87] (see Section 2.3.9 of Chapter 2 for a more detailed discussion of these methods). However for this simple class of models, these approximate methods are not necessary; with a slight reformulation of the two models, the Bayes factor can be calculated analytically and the need for MCMC sampling is eliminated.

5.2.2 Derivation of Analytic Expression for Distribution of Log-Fold-Change

The models described above can be reformulated slightly, exploiting conjugate prior distribution so that an analytic expression for the posterior distribution of the log-fold-change, given observed protein-level quantifications, can be derived. As above, the full model is:

$$P(y|\mu, \sigma^2) = N(\mu, \sigma^2) \tag{5.16}$$

$$P(z|\mu, d, \sigma^2) = N(\mu + d, \sigma^2) \tag{5.17}$$

$$P(\sigma^2) = \text{Scale-Inv-}\chi^2(\nu, \tau^2) \tag{5.18}$$

where y is a vector of n_y log-intensities from the control group, and z is a vector of n_z log-intensities from the treatment group.

This can be re-expressed in terms of $\mathbf{y} = (y, z)$ and $\boldsymbol{\mu} = (\mu, d)$:

$$P(\mathbf{y}|\boldsymbol{\mu}, \sigma^2) = MVN(\mathbf{X}\boldsymbol{\mu}, \mathbf{I}\sigma^2) \tag{5.19}$$

such that $\mathbf{X}$ is a $(n_y + n_z) \times 2$ design matrix:

$$\mathbf{X} = \begin{bmatrix} 1 & -n_y/(n_y + n_z) \\ 1 & -n_y/(n_y + n_z) \\ \vdots & \vdots \\ 1 & -n_y/(n_y + n_z) \\ 1 & n_z/(n_y + n_z) \\ 1 & n_z/(n_y + n_z) \\ \vdots & \vdots \\ 1 & n_z/(n_y + n_z) \end{bmatrix} \tag{5.20}$$

Then a multivariate normal-scale-inverse-χ^2 prior (a reparameterisation of the multivariate normal-inverse-gamma distribution with a scale-inverse-χ^2 prior on variance) can be assigned to $(\boldsymbol{\mu}, \sigma^2)$:

$$P(\boldsymbol{\mu}, \sigma^2) = \text{MVN-Scale-Inv-}\chi^2(\boldsymbol{\mu}_0, \mathbf{V}, \nu, \tau^2) \tag{5.21}$$

where $\boldsymbol{\mu}_0 = (\mu_0^\mu, \mu_0^d)$ is the prior mean and $\mathbf{V}$ is a diagonal matrix informing the prior belief of $\boldsymbol{\mu}_0$:

$$\mathbf{V} = \begin{bmatrix} \lambda_0^\mu & 0 \\ 0 & \lambda_0^d \end{bmatrix} \tag{5.22}$$

where λ_0^{μ} and λ_0^{d} inform the prior belief in the prior means of μ and d respectively. Since the multivariate normal-scale-inverse-χ^2 is a conjugate prior to the multivariate normal distribution, the parameters $\boldsymbol{\mu}_0$, $\mathbf{V}$, ν and τ^2 can be updated analytically to give $\boldsymbol{\mu}_0'$, $\mathbf{V}'$, ν' and $\tau^{2'}$ (see appendix B for further details):

$$P(\boldsymbol{\mu}, \sigma^2|\mathbf{y}) = \text{MVN-Scale-Inv-}\chi^2(\boldsymbol{\mu}_0', \mathbf{V}', \nu', \tau^{2'}) \tag{5.23}$$

The marginal posterior distribution of d is conveniently described by a student-t distribution:

$$P(d|\mathbf{y}, M_F) = \text{Student-T}(\nu', \mu_0^{d'}, \tau^{2'} \cdot \lambda_0^{d'}) \tag{5.24}$$

The marginal likelihood is also expressible analytically as the density of a multivariate Student-T distribution at the point $\mathbf{y}$:

$$P(\mathbf{y}|M_F) = \text{MV-Student-T}(\mathbf{y}|\nu, \mathbf{X}\boldsymbol{\mu}_0, \tau^2 \cdot (\mathbf{I} + \mathbf{X}\mathbf{V}\mathbf{X}^T)) \tag{5.25}$$

The marginal likelihood of the null model is derived similarly, meaning that the Bayes factor can be calculated *exactly* without need for estimation by taking the ratio of the two marginal likelihoods:

$$B_{N,F} = \frac{P(\mathbf{y}|M_N)}{P(\mathbf{y}|M_F)} \tag{5.26}$$

Hence, rapid Bayesian estimation of the posterior log-fold-change and resulting Bayes factor is achievable by choosing appropriate prior values and updating them conditional on observed protein log-intensities. The Bayes factor provides a measure of the goodness-of-fit of a model, but with the ability to show a preference for the null model; the Bayes factor penalises the number of parameters since the prior distribution in the full model is spread over a higher dimensional space[72].

5.3 Prior Parameters

Bayesian prior parameters by their very nature are subjective and, as such, should be chosen in such a way to reflect prior beliefs about the parameters in question. More importantly, Bayes factors can only be considered accurate when the models being

compared are realistic[72]; impossible values should be excluded by the prior. Here the prior parameters chosen for the Bayesian model comparison are justified.

5.3.1 Choice of μ_0^d

The choice of μ_0^d declares the prior belief in the mean log-fold-change for the protein. Since in any -omics experiment where differential expression is being tested for, it is generally expected that the majority of proteins (or genes, transcripts or metabolites) are unchanging between test conditions. Hence, the most (and arguably the only) sensible choice of μ_0^d is $\mu_0^d = 0$.

5.3.2 Residual Variance

Gelman et al. [106] specifically make recommendations for priors on variance in hierarchical models. In particular, they advise against the use of inverse-gamma priors of the form Inv-Gamma(ϵ, ϵ) for small ϵ, observing that priors of this form are inappropriate for data where small values of σ^2 are possible, since in these scenarios inference is sensitive to the value of ϵ chosen.

Instead, the empirical Bayes method used in limma[40][120] is employed to estimate the parameters of a hyperprior of variance across the population of proteins from the observed per-protein sample variances, calculated from the observed protein-level quantifications. For a single protein j with estimated sample quantifications $y_{1:N}$, the sample variance is calculated as:

$$\sum_{i=1}^{N} (y_{i,j} - \bar{\mathbf{y}}_\mathbf{j})^2 \tag{5.27}$$

These are then used by limma to estimate the ν_0 and τ_0^2 parameters on variance.

For example, for the single fraction spike-in data tested in Chapter 3 for one of the BayesProt runs, ν_0 and τ_0^2 are estimated to be $\nu_0 = 2.14$ and $\tau_0^2 = 0.0169$. This is, of course, an approximation to an idealised hierarchical model where a hyper-prior is applied to the variance across all proteins.

5.3.3 Choice of λ_0^d — Implicit Prior on Log-fold-change

The choice of λ_0^d is important, since this effectively declares the prior strength of belief in the prior mean log-fold-change $\mu_0^d = 0$. The choice of prior parameters can also be

interpreted in terms of their effect on the implicit prior on log-fold-change, that is, the marginal prior distribution of d:

$$P(d) = \int_{\sigma^2} \int_{\mu} P(d|\mu, \sigma^2) \tag{5.28}$$

This marginal prior distribution takes the form of a scaled, non-centred Student-T distribution:

$$P(d) = \text{Student-T}\left(\nu_0, \mu_0^d, \sqrt{\frac{\tau_0^2}{\lambda_0^d}}\right) \tag{5.29}$$

with degrees of freedom ν_0, location parameter μ_0^d and scaling parameter $\sqrt{\frac{\tau_0^2}{\lambda_0^d}}$. Setting $\lambda_0^d = 1$ would immediately seem to be a reasonable default; however, this has the side effect of undesirably shrinking posterior estimates of log-fold-change towards zero: higher values of λ_0^d imply a stronger belief in the prior of the mean μ_d^0.

In the implicit prior in (5.29), the term $\sqrt{\frac{\tau_0^2}{\lambda_0^d}}$ determines the scaling of the distribution; in the case where $\nu = \infty$ and the Student-T is equivalent to a normal distribution, this term would be equal to the standard deviation of the distribution.

As discussed in [72] and mentioned above, the choice of a sensible prior is essential to Bayesian model comparison; the prior must not exclude reasonable values, but also should exclude unreasonable ones. For example, it has been shown that the number of copies of any protein in a human cell ranges from a few copies per cell up to 20,000,000 copies per cell [121]. Hence the maximum possible fold-change is around 1×10^7, which corresponds to a $\log_2$-fold-change of around 24. The prior on $\log_2$ fold-change should allow for the possibility of $\log_2$-fold-changes of ± 24 but exclude "impossible" values such as 1000.

Let σ_d be the scale parameter of the implicit prior. Then λ_0^d can be chosen indirectly by inferring its value from the estimated τ_0^2 and the desired scale of the implicit prior

on fold-change, σ_d:

$$\sigma_d = \sqrt{\frac{\tau_0^2}{\lambda_0^d}} \tag{5.30}$$

$$\implies \lambda_0^d = \frac{\tau_0^2}{\sigma_d^2} \tag{5.31}$$

The effect of varying σ_d was investigated by applying the Bayesian model comparison to the single fraction spike-in data used for validation of the BayesProt model in Chapter 3. The resulting implicit priors on log-fold-change are presented in Figure 5.1. The resulting precision–recall curves are presented in Figure 5.2.

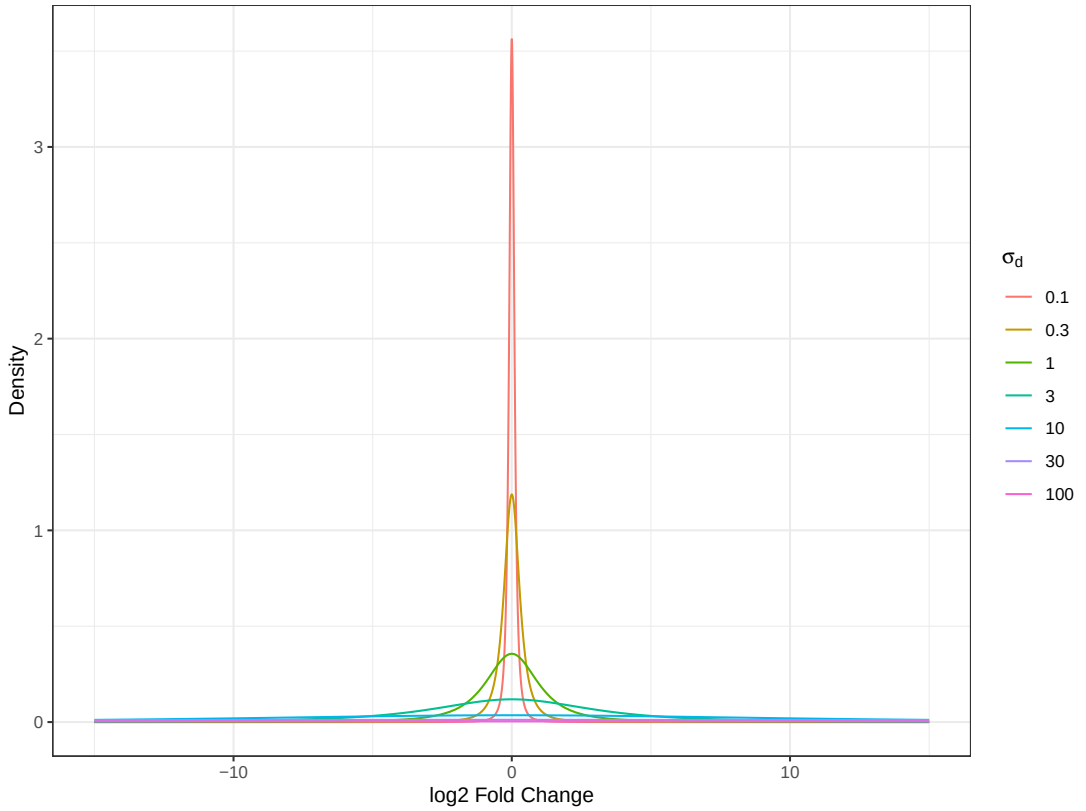

Figure 5.1: Example of implicit priors on $\log_2$-fold-change for one run of the spike-in data considered in Chapter 3, student-T distributions with degrees of freedom parameter $\nu_0 = 2.14$ and a varying scale parameter σ_d.

In general, increasing the variance of the implicit fold-change prior (which corresponds to decreasing λ_0^d) results in FDR estimates becoming more conservative.

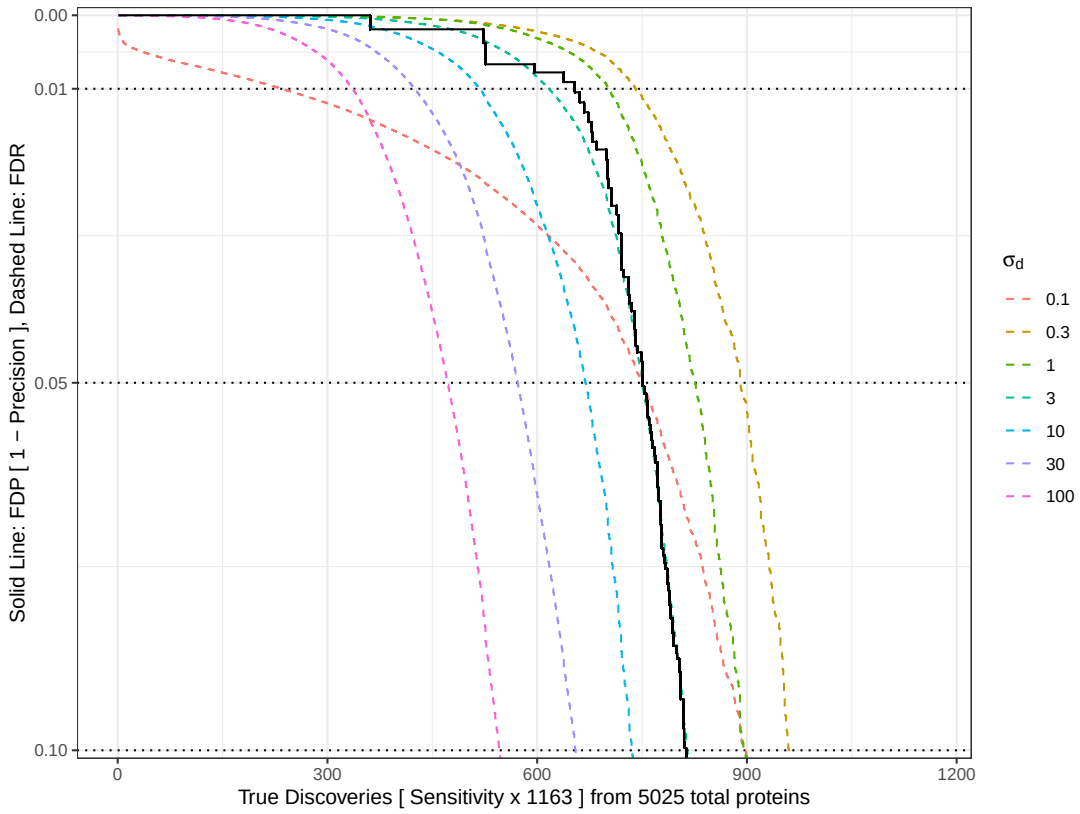

Figure 5.2: Precision–recall curves showing the effect of varying σ_d on an example from the spike-in data set used for validation in Chapter 3. Estimated FDRs for varying values of the σ_d are shown in different colours. The FDP is unchanged by varying the scale parameter and is shown in black.

For the purposes of validating the Bayesian model comparison against other methods in Section 5.5, the value of $\sigma_d = 10$ was chosen; this results in conservative estimates of FDR, and is the most comparable to the prior distribution on log-fold-change used in QPROT (Normal$(0, 10)$). The same prior on $\log_2$-fold-change was used for the MCMC-based tests.

5.3.4 Choice of μ_0^μ and λ_0^μ

It should be noted that the choice of μ_0^μ and λ_0^μ has no effect on the resulting Bayes factor: since these parameters appear in both the null and full models, their effect is essentially removed when comparing the two models. The values $\mu_0^\mu = 0$ and $\lambda_0^\mu = \lambda_0^d$ are chosen arbitrarily.

5.4 Incorporating Uncertainty

Chapter 3 demonstrated that the propagation of the uncertainty in protein quantification estimates is essential to achieve accurate estimates of differential expression. For any given protein, BayesProt outputs estimates of the medians $y_{1:K}$ and robust estimates of the standard deviations (median absolute deviations) $\mathrm{SD}_{1:K}$ of the K assay effects. In a full hierarchical model it would be more appropriate to handle the uncertainty estimates by including SD in the likelihood function:

$$y_i \sim \text{Normal}\left(\theta_i, \sqrt{\sigma^2 + \mathrm{SD}_i^2}\right) \quad \text{for } i = 1 : K \tag{5.32}$$

where θ_i is the latent predictor for assay i for that protein.

However, the resulting posterior is not amenable to conjugate analysis with the model described above and would necessitate numerical integration through, for example, MCMC. Additionally, loss of conjugacy would mean that the calculation of the marginal likelihoods (and therefore Bayes factors and posterior error probabilities) is no longer analytic.

5.4.1 Per-Protein Informative Prior on σ

In order to maintain conjugacy, an approximate method is adopted by adjusting the prior on residual variance on a per-protein basis to incorporate the estimated uncertainties $\mathrm{SD}_{1:K}$. The uncertainty in the protein quantification estimates can be concep-

tualised as prior knowledge used to inform the prior distribution on σ for each protein separately, updating the prior in a Bayesian fashion:

$$\rho(\sigma^2) = \text{Scale-Inv-}\chi^2\left(\nu = \nu_0 + K, \tau^2 = \frac{\nu_0\tau_0^2 + \sum_{i=1}^{K}(\text{SD}_i^2)}{\nu_0 + K}\right) \tag{5.33}$$

The result is that proteins with greater uncertainty in their quantifications (and therefore larger values for SD) have their prior variances inflated.

5.5 Methods

5.5.1 Data

Spike-in Data

For validation we utilise the same three spike-in data sets that were used to validate the BayesProt algorithm in Chapter 3, using the protein quantification estimates generated by BayesProt with the full empirical Bayes approach.

For each of the three spike-in data sets, as in Chapter 3, the protein quantifications output by BayesProt were normalised using only the background rat proteins. The fold-changes represented in these data sets are largely in one direction due to the spike-in nature of the experiment. However, this skewed distribution of fold-changes does not realistically represent the distribution of fold-changes that would be expected in a typical clinical data set.

Data with Simulated Fold-change

In order to address this shortcoming, a number of additional data sets were generated using the spike-in data as a basis, taking the protein-level quantification estimates previously generated by BayesProt with the full empirical Bayes approach and artificially inducing known fold-changes to some of the proteins. Firstly, all non-rat proteins were removed from the data, so that only the set of rat proteins with no fold-change remained but with a realistic distribution of error, which is the sum of technical variation and the error due to MCMC sampling. Furthermore, the distribution of number of peptides across all rat proteins is typical of a biological data set.

The proteins were shuffled randomly, before assigning artificial fold-changes to the latter 50% of the list. Multiple simulated data sets were created, in each of

which artificial fold-changes were randomly sampled from a different distribution. The distributions used were as follows: Normal$(0, 0.25)$, Normal$(0, 0.5)$, Normal$(0, 1.0)$, Normal$(0.5, 0.25)$ and a bimodal Normal$(\pm 0.5, 0.25)$.

The same "true" fold-change for each differentially expressed protein is induced in each of the replicate runs of BayesProt. The induced fold-changes were also given some normally distributed error, Normal$(0.0, 0.5)$ for each of the four assays in the second group.

While these distributions represent relatively small ranges of fold-changes when compared to spike-in data sets, the data sets which are created are more challenging than a conventional spike-in experiment with large fold-changes largely in one direction.

5.5.2 Comparison with Other Techniques

The results of the Bayesian model comparison test are compared with those results generated by a number of other techniques for differential expression testing including, for the purposes of clarity and brevity, a selection of those techniques already benchmarked in Chapter 3.

Further to these previously compared methods, versions of the MCMC-based tests are included that utilise the same limma-derived empirical Bayes prior on residual variance that is used for the Bayesian model comparison.

Additionally, the QPROT[43] software was tested using the median protein quantification estimates generated by BayesProt as input data. The QPROT software utilises a similar Bayesian model to both the Bayesian t-tests and the Bayesian model comparison but uses MCMC to compute Z-statistics before estimating FDR via a mixture model. For each data set and run of BayesProt quantifications, QPROT was run with 10,000 warmup iterations and 100,000 sampling iterations (the values used in [43]) with the normalisation disabled, since the data was manually normalised to the rat background.

False discovery rate estimation was also performed using a number of techniques to be compared with the Bayesian model comparison: as in Chapter 3 to calculate FDRs for the metafor t-test and MCMC-based tests the ash R package was used in four configurations: using t-statistics with per-protein degrees of freedom and a uniform mixture component distribution (A); using Z-statistics and a uniform mixture component distribution (A0); using t-statistics with per-protein degrees of freedom and a half-uniform mixture component distribution (AH); and using Z-statistics and a half-uniform mixture component distribution (AH0). QPROT's FDR estimation is applied to the results from QPROT, and BMC's FDR was calculated as the cumulative

mean of posterior error probabilities. Approximate calculation of Bayes factors from the MCMC-based tests was achieved by applying the Savage–Dickey density ratio[91] (SDDR) so that the effect of an approximation of the Bayes factor could be compared with the analytic method outlined in Section 5.2.2. The density of the posterior distribution of the $\log_2$-fold-change was approximated by taking the mean and variance of the posterior samples. The density of this approximate normal posterior was then compared with the density of the prior Normal$(0, 10)$ at $d = 0$ to calculate an approximation of the Bayes factor and subsequently the posterior error probability for each protein in the same manner as for the analytic Bayesian model comparison.

The methods being compared are summarised in Table 5.1.

Table 5.1: Summary of methods compared in this chapter.

Code	Quantification Method	FDR Estimation Method
BP+BMC	BMC	BMC
BP+QPROT	QPROT	QPROT
BP+MCMCS0+A	MCMC (equal vars, no SE)	ash (t-statistic, uniform)
BP+MCMCS0+A0	MCMC (equal vars, no SE)	ash (Z-statistic, uniform)
BP+MCMCS0+AH	MCMC (equal vars, no SE)	ash (t-statistic, half-uniform)
BP+MCMCS0+AH0	MCMC (equal vars, no SE)	ash (Z-statistic, half-uniform)
BP+MCMCS0+SDDR	MCMC (equal vars, no SE)	Savage–Dickey density ratio
BP+M+A	Metafor	ash (t-statistic, uniform)
BP+M+A0	Metafor	ash (Z-statistic, uniform)
BP+M+AH	Metafor	ash (t-statistic, half-uniform)
BP+M+AH0	Metafor	ash (Z-statistic, half-uniform)
BP+MCMCS+A	MCMC (equal vars)	ash (t-statistic, uniform)
BP+MCMCS+A0	MCMC (equal vars)	ash (Z-statistic, uniform)
BP+MCMCS+AH	MCMC (equal vars)	ash (t-statistic, half-uniform)
BP+MCMCS+AH0	MCMC (equal vars)	ash (Z-statistic, half-uniform)
BP+MCMCS+SDDR	MCMC (equal vars)	Savage–Dickey density ratio
BP+MCMCW+A	MCMC (unequal vars)	ash (t-statistic, uniform)
BP+MCMCW+A0	MCMC (unequal vars)	ash (Z-statistic, uniform)
BP+MCMCW+AH	MCMC (unequal vars)	ash (t-statistic, half-uniform)
BP+MCMCW+AH0	MCMC (unequal vars)	ash (Z-statistic, half-uniform)
BP+MCMCW+SDDR	MCMC (unequal vars)	Savage–Dickey density ratio
BP+MCMCSEB+A	MCMC (equal vars, LIMMA EB)	ash (t-statistic, uniform)
BP+MCMCSEB+A0	MCMC (equal vars, LIMMA EB)	ash (Z-statistic, uniform)
BP+MCMCSEB+AH	MCMC (equal vars, LIMMA EB)	ash (t-statistic, half-uniform)
BP+MCMCSEB+AH0	MCMC (equal vars, LIMMA EB)	ash (Z-statistic, half-uniform)
BP+MCMCSEB+SDDR	MCMC (equal vars, LIMMA EB)	Savage–Dickey density ratio
BP+MCMCWEB+A	MCMC (unequal vars, LIMMA EB)	ash (t-statistic, uniform)
BP+MCMCWEB+A0	MCMC (unequal vars, LIMMA EB)	ash (Z-statistic, uniform)
BP+MCMCWEB+AH	MCMC (unequal vars, LIMMA EB)	ash (t-statistic, half-uniform)
BP+MCMCWEB+AH0	MCMC (unequal vars, LIMMA EB)	ash (Z-statistic, half-uniform)
BP+MCMCWEB+SDDR	MCMC (unequal vars, LIMMA EB)	Savage–Dickey density ratio

5.6 Results

For each of the data sets, a set of precision–recall curves and set of FDP vs FDR curves are shown, with curves separated by quantification method across columns and method for FDR estimation across rows into separate panels within each plot for clarity. The curves are also coloured according to the quantification method used.

The y-axes of the precision–recall plots and both the x and y axes of the FDR-FDP plots are truncated at an FDR/FDP of 0.1; this region covers the range of commonly used cutoffs for FDR in many studies.

Alongside these figures, as in Section 3.5, tables showing a metric of calibrated recall are presented. For each method the mean recall (and standard deviation of recall) across the replicates is calculated at each of three commonly used FDR thresholds, 1%, 5% and 10%. Values are only shown for methods where ground-truth FDP is consistently less than the estimated FDR (i.e. the FDR is calibrated) at that FDR cutoff. This serves as a summary metric to determine the usefulness of the FDR estimation methods; while some methods are able to achieve better recall at a given FDP, in many real experiments the ground truth cannot be known, and so it is preferable that methods should be able to give calibrated FDRs.

5.6.1 Spike-in Data — Single Fraction

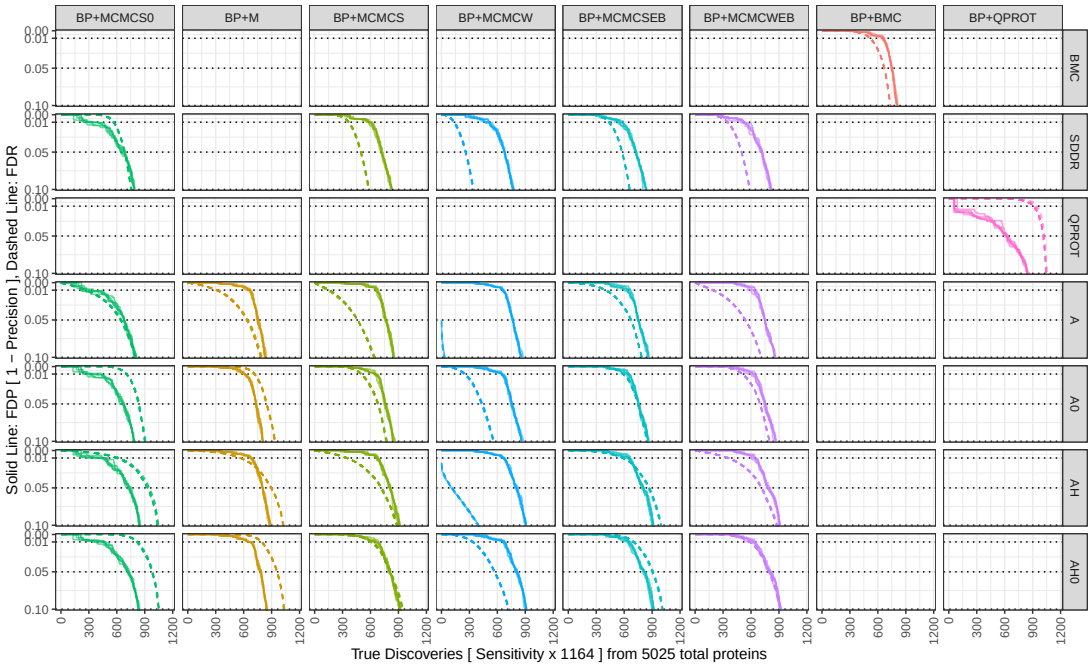

Figure 5.3: Precision–recall curves for multiple methods for the spike-in single-fraction data. Curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method. FDP curves are shown with solid lines, and corresponding FDR estimates are shown with dotted lines. Only points with FDR/FDP < 0.1 are shown. On these plots, methods whose FDR curve is *below* the FDP curve can be considered to be calibrated.

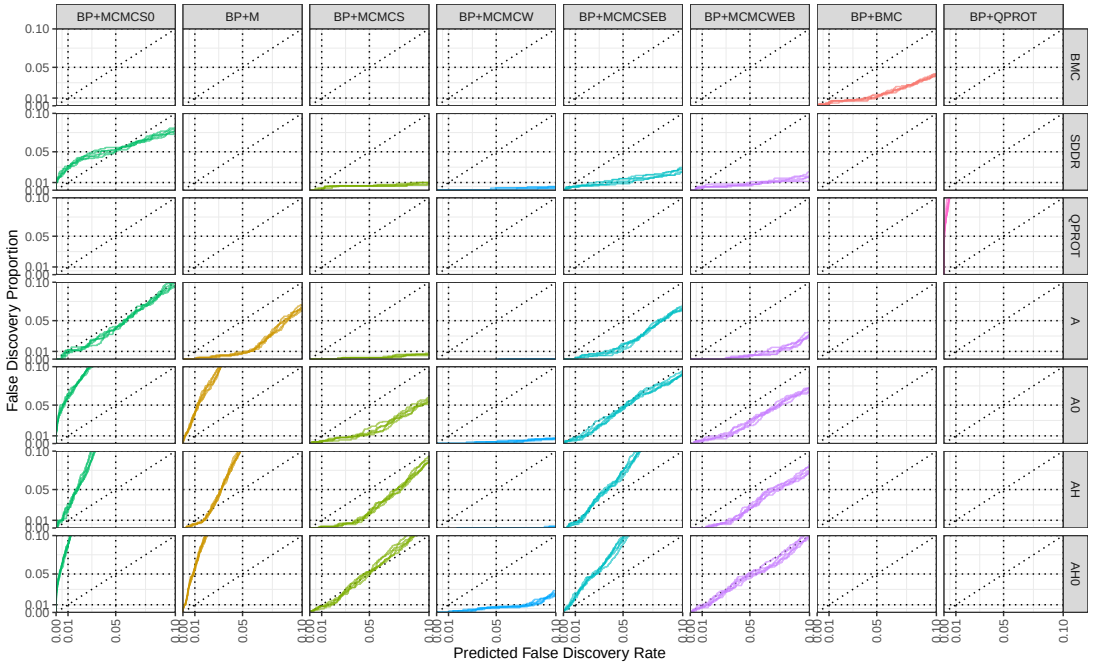

Figure 5.4: FDP vs FDR curves for multiple methods for the spike-in single-fraction data. Curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method.

Table 5.2: Calibrated recall values on the single-fraction spike-in data for all methods tested at multiple FDR cutoffs.

	Calibrated Recall		
Method	1% FDR	5% FDR	10% FDR
BP+BMC	517.6 ± 1.14	668.6 ± 2.3	733.8 ± 1.48
BP+QPROT			
BP+MCMCS0+A		643.2 ± 5.31	
BP+MCMCS0+A0			
BP+MCMCS0+AH			
BP+MCMCS0+AH0			
BP+MCMCS0+SDDR			753.2 ± 1.3
BP+M+A	309.8 ± 5.85	637.6 ± 6.54	791.2 ± 5.76
BP+M+A0			
BP+M+AH	565.4 ± 1.14		
BP+M+AH0			
BP+MCMCS+A	153.2 ± 9.15	461 ± 2.12	641.2 ± 4.76
BP+MCMCS+A0	501.8 ± 4.02	680 ± 3.87	772 ± 5.39
BP+MCMCS+AH	346.4 ± 4.28	$\mathbf{714.6 \pm 4.62}$	$\mathbf{888.8 \pm 6.57}$
BP+MCMCS+AH0	$\mathbf{607.8 \pm 6.69}$		
BP+MCMCS+SDDR	366.2 ± 1.64	497 ± 3.24	583.2 ± 4.76
BP+MCMCW+A			33.8 ± 2.68
BP+MCMCW+A0	249.8 ± 3.56	435 ± 4.42	560 ± 3.39
BP+MCMCW+AH		110.6 ± 3.36	394.6 ± 4.93
BP+MCMCW+AH0	337.6 ± 4.04	571.2 ± 2.59	722.6 ± 5.5
BP+MCMCW+SDDR	136 ± 3.46	259.8 ± 2.95	343 ± 3.94
BP+MCMCSEB+A	398.8 ± 5.63	671.8 ± 3.03	790.2 ± 1.48
BP+MCMCSEB+A0	$\mathbf{589.2 \pm 3.27}$	$\mathbf{748.4 \pm 1.67}$	$\mathbf{847.2 \pm 1.92}$
BP+MCMCSEB+AH	587.8 ± 5.07		
BP+MCMCSEB+AH0			
BP+MCMCSEB+SDDR	438.4 ± 3.05	577.2 ± 1.1	658.6 ± 3.21
BP+MCMCWEB+A	189.2 ± 5.4	537.4 ± 2.97	707.8 ± 1.64
BP+MCMCWEB+A0	508.6 ± 2.41	696.2 ± 3.11	798.8 ± 1.3
BP+MCMCWEB+AH	353 ± 5.43	$\mathbf{712 \pm 3.39}$	$\mathbf{877.2 \pm 4.66}$
BP+MCMCWEB+AH0	$\mathbf{588.4 \pm 2.07}$		
BP+MCMCWEB+SDDR	348.4 ± 2.7	498.4 ± 1.52	587 ± 4.3

For those methods using BayesProt input data, mean recalls across five replicated BayesProt runs are shown, along with the estimated standard deviation. The values for the three best-performing methods in each column are highlighted in bold for emphasis.

5.6.2 Spike-in Data — Pooled Fractions

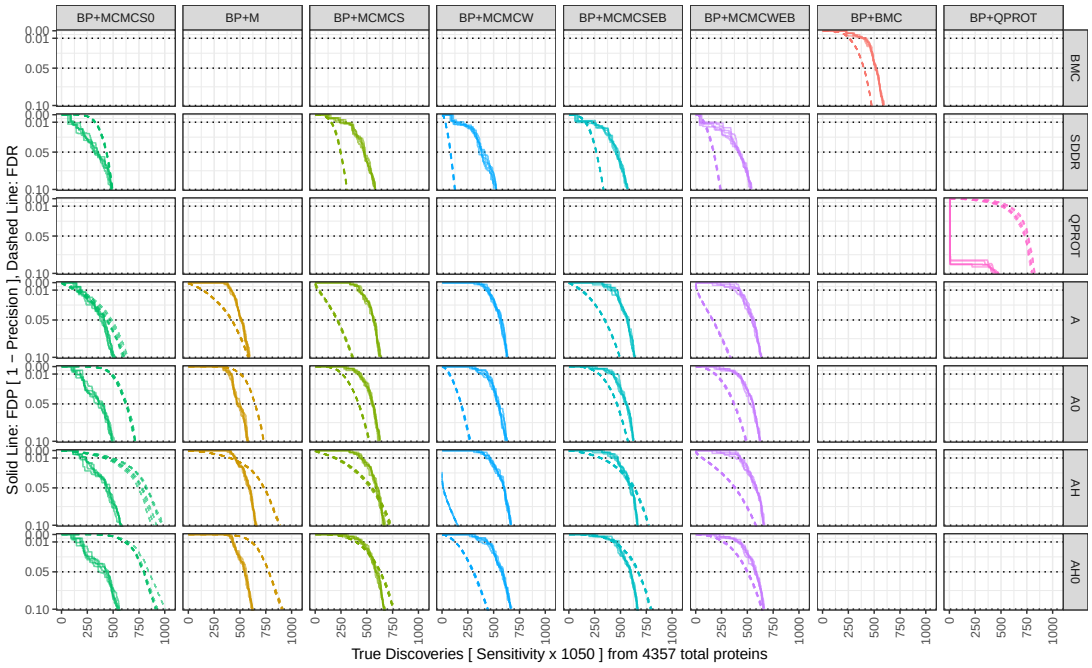

Figure 5.5: Precision–recall curves for the pooled fraction spike-in data. As above, curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method. FDP curves are shown with solid lines, and corresponding FDR estimates are shown with dotted lines. Only points with FDR/FDP < 0.1 are shown. On these plots, methods whose FDR curve is *below* the FDP curve can be considered to be calibrated.

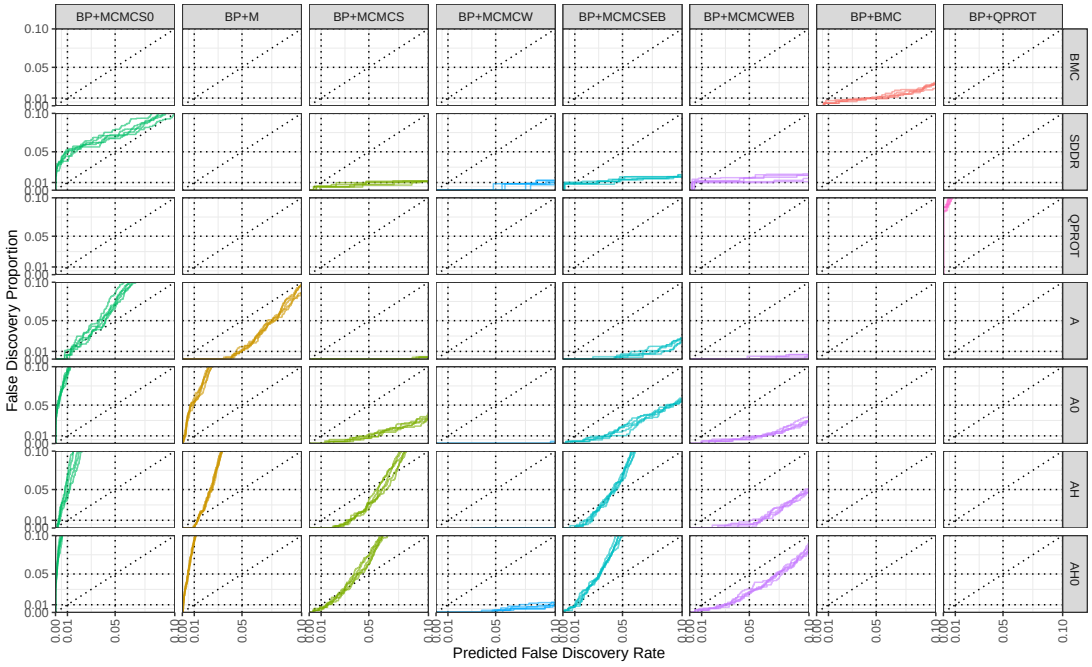

Figure 5.6: FDP vs FDR curves for multiple methods for the spike-in pooled fraction data. Curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method.

Table 5.3: Calibrated recall values for the pooled fraction spike-in data for all methods tested at multiple FDR cutoffs.

	Calibrated Recall		
Method	1% FDR	5% FDR	10% FDR
BP+BMC	256.8 ± 0.837	400.2 ± 1.1	475.4 ± 1.14
BP+QPROT			
BP+MCMCS0+A			
BP+MCMCS0+A0			
BP+MCMCS0+AH			
BP+MCMCS0+AH0			
BP+MCMCS0+SDDR			
BP+M+A	137.8 ± 2.28	419.6 ± 2.79	$\mathbf{581.2 \pm 3.27}$
BP+M+A0			
BP+M+AH	$\mathbf{403.6 \pm 3.36}$		
BP+M+AH0			
BP+MCMCS+A	32.6 ± 7.7	214.2 ± 4.76	362.6 ± 5.13
BP+MCMCS+A0	260.8 ± 2.17	421.4 ± 3.05	524.8 ± 3.27
BP+MCMCS+AH	192.8 ± 7.85	$\mathbf{545.4 \pm 10.4}$	
BP+MCMCS+AH0	$\mathbf{393.8 \pm 0.447}$		
BP+MCMCS+SDDR	154.6 ± 2.97	246 ± 2.55	309.8 ± 2.17
BP+MCMCW+A			
BP+MCMCW+A0	71.6 ± 3.65	184.6 ± 3.65	272.4 ± 5.08
BP+MCMCW+AH		11.8 ± 2.49	150.8 ± 2.05
BP+MCMCW+AH0	126.4 ± 2.3	303.2 ± 5.76	441.6 ± 3.51
BP+MCMCW+SDDR	31.8 ± 1.48	81.8 ± 2.68	124.2 ± 1.92
BP+MCMCSEB+A	117.4 ± 7.23	346.2 ± 3.27	493 ± 2.74
BP+MCMCSEB+A0	291.2 ± 1.64	$\mathbf{465.6 \pm 3.36}$	572 ± 3
BP+MCMCSEB+AH	307.4 ± 3.85		
BP+MCMCSEB+AH0	$\mathbf{437 \pm 4.47}$		
BP+MCMCSEB+SDDR		265.4 ± 2.07	335.6 ± 2.19
BP+MCMCWEB+A	11.6 ± 2.3	164.4 ± 3.05	333.4 ± 5.03
BP+MCMCWEB+A0	188.4 ± 4.62	371.6 ± 3.65	489.8 ± 5.45
BP+MCMCWEB+AH	63.6 ± 3.36	381 ± 4.53	$\mathbf{582.4 \pm 3.05}$
BP+MCMCWEB+AH0	269.4 ± 6.54	$\mathbf{500.8 \pm 5.07}$	$\mathbf{641 \pm 1.87}$
BP+MCMCWEB+SDDR		173 ± 3.54	240.2 ± 4.09

For those methods using BayesProt input data, mean recalls across five replicated BayesProt runs are shown, along with the estimated standard deviation. The values for the three best-performing methods in each column are highlighted in bold for emphasis.

5.6.3 Spike-in Data — Faulty Data

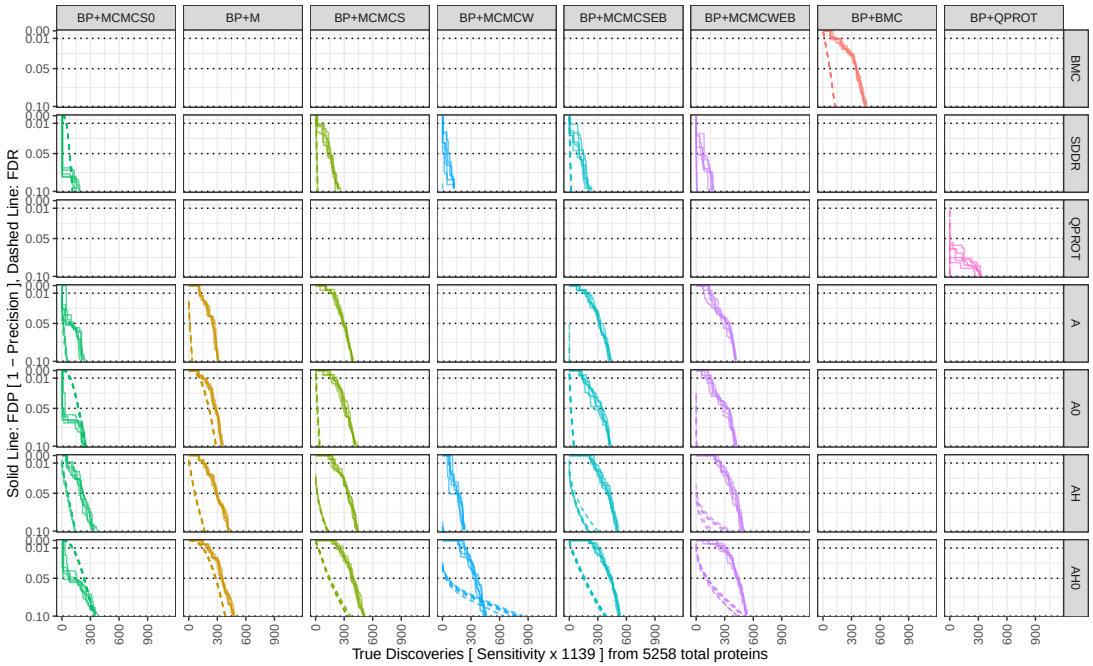

Figure 5.7: Precision–recall curves for the faulty spike-in data. As above, curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method. FDP curves are shown with solid lines, and corresponding FDR estimates are shown with dotted lines. Only points with FDR/FDP < 0.1 are shown. On these plots, methods whose FDR curve is *below* the FDP curve can be considered to be calibrated.

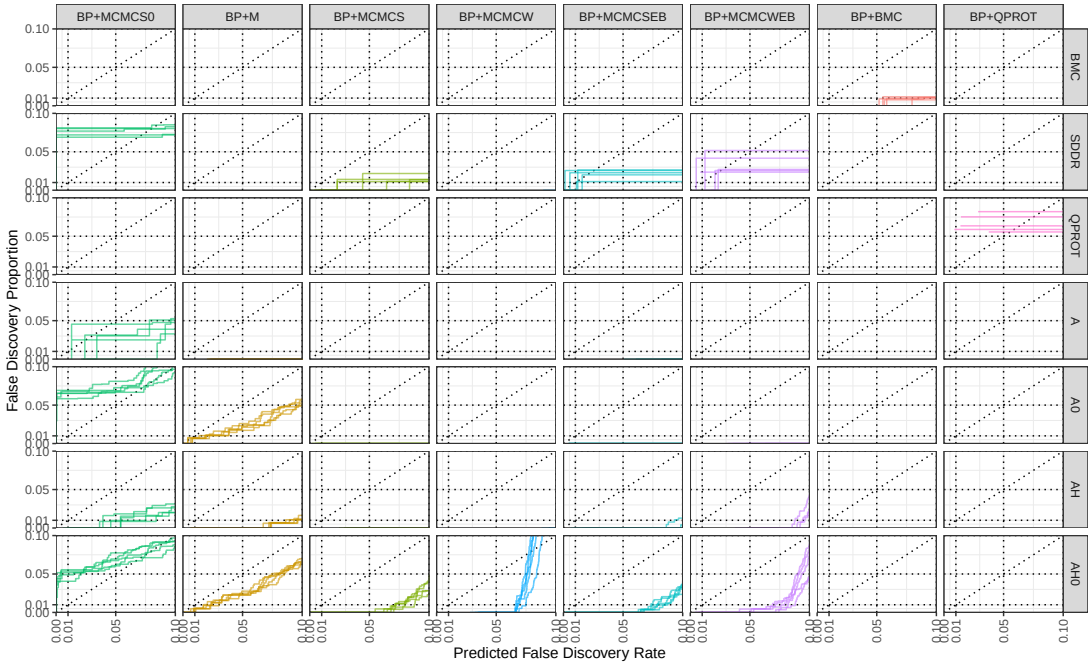

Figure 5.8: FDP vs FDR curves for multiple methods for the faulty spike-in data. Curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method.

Table 5.4: Calibrated recall values for the faulty spike-in data for all methods tested at multiple FDR cutoffs.

	Calibrated Recall		
Method	1% FDR	5% FDR	10% FDR
BP+BMC	15 ± 0.707	71.2 ± 0.837	126 ± 0.707
BP+QPROT			1.8 ± 2.17
BP+MCMCS0+A		13.2 ± 5.4	51.8 ± 5.72
BP+MCMCS0+A0			
BP+MCMCS0+AH		56.2 ± 6.3	141.8 ± 4.6
BP+MCMCS0+AH0			334.4 ± 3.29
BP+MCMCS0+SDDR			110.8 ± 2.49
BP+M+BH			
BP+M+S			
BP+M+A		10.4 ± 1.14	35.4 ± 0.548
BP+M+A0	$\mathbf{113.6 \pm 1.14}$	$\mathbf{211.8 \pm 1.92}$	286.6 ± 1.82
BP+M+AH	4 ± 0.707	63.6 ± 2.07	166.4 ± 3.13
BP+M+AH0	$\mathbf{147.2 \pm 1.92}$	$\mathbf{282.4 \pm 2.51}$	$\mathbf{382.8 \pm 1.48}$
BP+MCMCS+A			
BP+MCMCS+A0	3.8 ± 1.92	19.4 ± 2.88	41 ± 3.16
BP+MCMCS+AH		17.4 ± 2.3	126.6 ± 4.39
BP+MCMCS+AH0	$\mathbf{26.8 \pm 2.86}$	143.6 ± 5.41	348.2 ± 16
BP+MCMCS+SDDR	2.6 ± 1.14	11.4 ± 2.19	22.2 ± 2.17
BP+MCMCW+A			
BP+MCMCW+A0			
BP+MCMCW+AH			8.6 ± 6.8
BP+MCMCW+AH0		47 ± 9.82	
BP+MCMCW+SDDR			
BP+MCMCSEB+A			
BP+MCMCSEB+A0	3 ± 0.707	17.8 ± 1.48	48 ± 4.06
BP+MCMCSEB+AH		41.2 ± 11.6	225.2 ± 43.2
BP+MCMCSEB+AH0	17 ± 2.83	$\mathbf{156 \pm 6.16}$	$\mathbf{376 \pm 10.2}$
BP+MCMCSEB+SDDR		9.6 ± 1.14	20.2 ± 1.64
BP+MCMCWEB+A			
BP+MCMCWEB+A0			4 ± 1.41
BP+MCMCWEB+AH			268.6 ± 100
BP+MCMCWEB+AH0	2.4 ± 0.548	109.2 ± 11.6	$\mathbf{474.4 \pm 39.4}$
BP+MCMCWEB+SDDR			7.6 ± 0.548

For those methods using BayesProt input data, mean recalls across five replicated BayesProt runs are shown, along with the estimated standard deviation. The values for the three best-performing methods in each column are highlighted in bold for emphasis.

5.6.4 Simulated Data — Normal$(0, 0.25)$

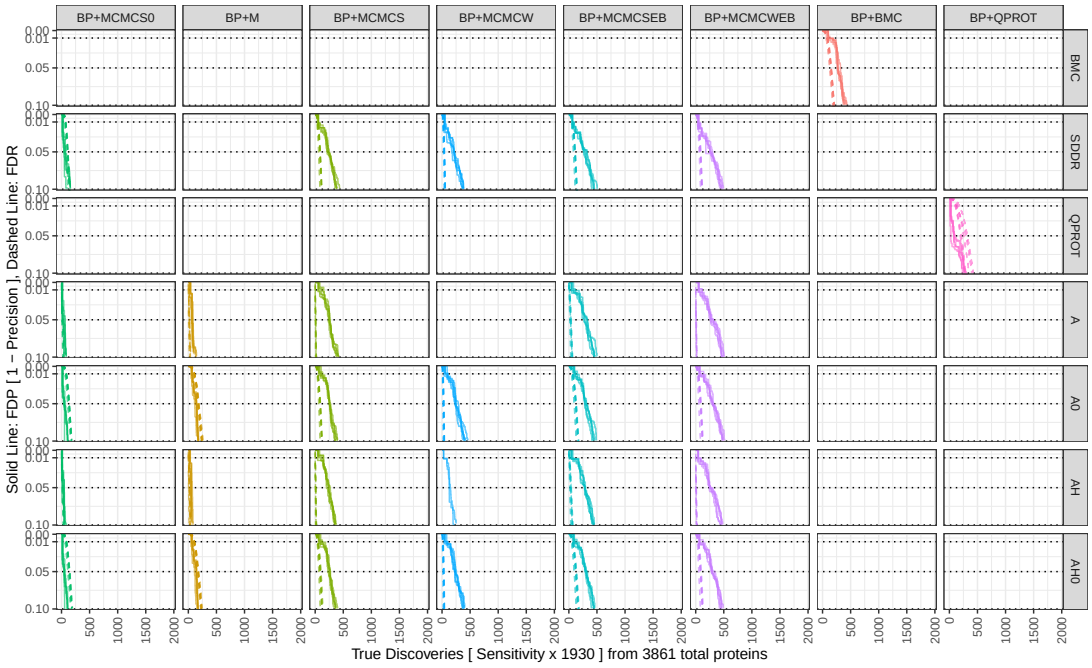

Figure 5.9: Precision-recall curves for the simulated data with fold-changes drawn from Normal$(0, 0.25)$. As above, curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method. FDP curves are shown with solid lines and corresponding FDR estimates are shown with dotted lines. Only points with FDR/FDP < 0.1 are shown. On these plots, methods whose FDR curve is *below* the FDP curve can be considered to be calibrated.

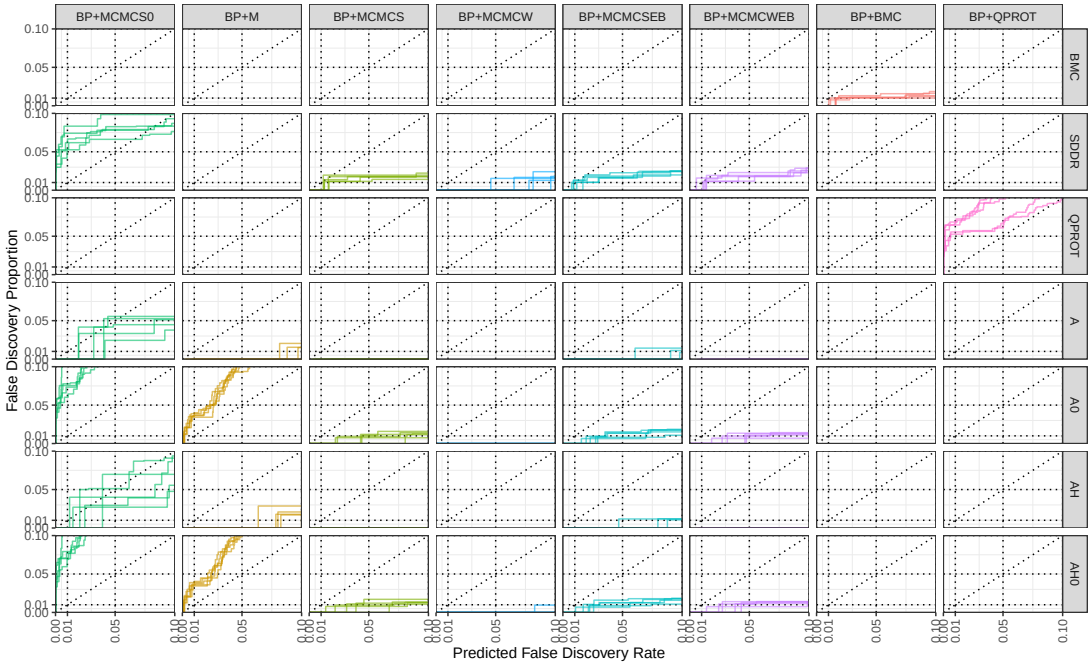

Figure 5.10: FDP vs FDR curves for the simulated data with fold-changes drawn from Normal$(0, 0.25)$. Curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method.

Table 5.5: Calibrated recall values for the simulated data with fold-changes drawn from Normal$(0, 0.25)$

	Calibrated Recall		
Method	1% FDR	5% FDR	10% FDR
BP+BMC	**81.6 ± 10.9**	**147 ± 12.2**	**198.8 ± 12.3**
BP+QPROT			
BP+MCMCS0+A			40.4 ± 11.7
BP+MCMCS0+A0			
BP+MCMCS0+AH	3.6 ± 3.44		45.2 ± 11.3
BP+MCMCS0+AH0			
BP+MCMCS0+SDDR			136.2 ± 12.2
BP+M+A		15.4 ± 6.54	32.6 ± 10.1
BP+M+A0			
BP+M+AH		14.4 ± 6.02	30.4 ± 10
BP+M+AH0			
BP+MCMCS+A		6.8 ± 3.7	16.2 ± 6.38
BP+MCMCS+A0	45.6 ± 7.4	83.4 ± 12	113.4 ± 14.9
BP+MCMCS+AH		8 ± 3.54	17.4 ± 6.47
BP+MCMCS+AH0	47 ± 8.09	86.8 ± 14.1	118 ± 17.1
BP+MCMCS+SDDR	47.2 ± 6.76	81.4 ± 11.7	108.2 ± 14
BP+MCMCW+A			
BP+MCMCW+A0	14.6 ± 5.32	26.6 ± 7.13	36.2 ± 8.96
BP+MCMCW+AH			
BP+MCMCW+AH0	14.4 ± 5.03	27.2 ± 7.43	36.8 ± 9.2
BP+MCMCW+SDDR	21.2 ± 6.02	38.2 ± 7.09	52.4 ± 8.76
BP+MCMCSEB+A	9 ± 4.58	31 ± 10	54 ± 12.6
BP+MCMCSEB+A0	**64.4 ± 10.5**	**118.2 ± 15.3**	**159.2 ± 16.3**
BP+MCMCSEB+AH	9 ± 4.58	32.2 ± 10.9	56.8 ± 14.8
BP+MCMCSEB+AH0	**65 ± 11.1**	**119.4 ± 16.6**	**161.6 ± 18.1**
BP+MCMCSEB+SDDR		103.2 ± 13.3	135.2 ± 14.8
BP+MCMCWEB+A		6.6 ± 4.16	17.2 ± 5.12
BP+MCMCWEB+A0	42.2 ± 8.61	81.2 ± 16.4	115.2 ± 18.5
BP+MCMCWEB+AH		6.8 ± 3.56	17.6 ± 5.86
BP+MCMCWEB+AH0	42.8 ± 9.04	82 ± 17	115.6 ± 18.7
BP+MCMCWEB+SDDR		80 ± 11.5	107.2 ± 12.9

For those methods using BayesProt input data, mean recalls across five replicated BayesProt runs are shown, along with the estimated standard deviation. The values for the three best-performing methods in each column are highlighted in bold for emphasis.

5.6.5 Simulated Data — Normal$(0, 0.5)$

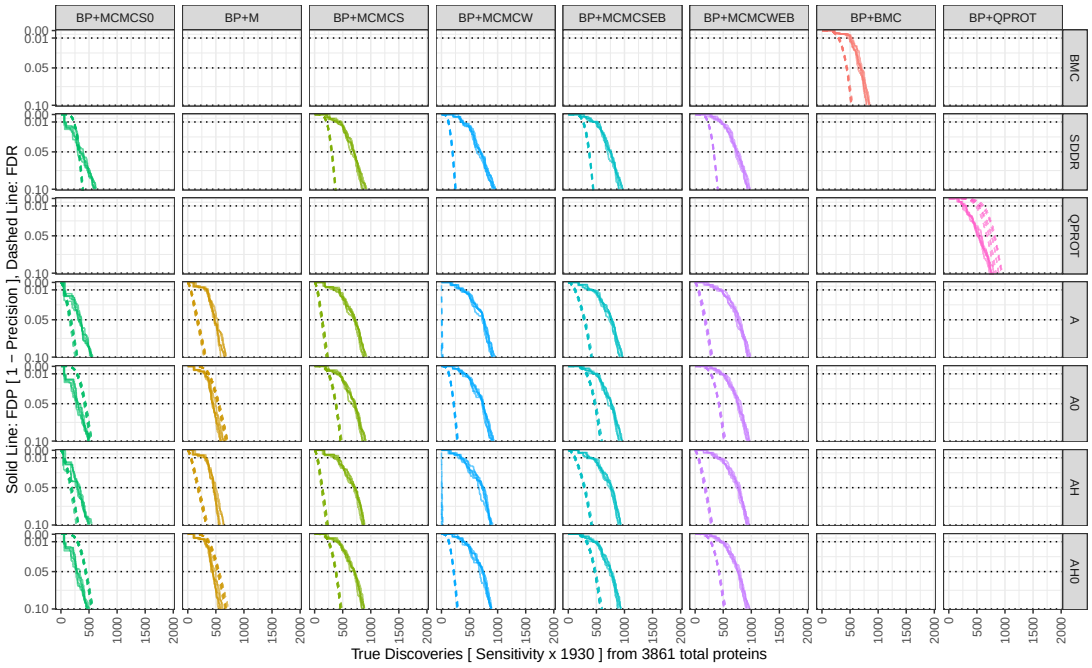

Figure 5.11: Precision–recall curves for multiple methods for the simulated data with fold-changes drawn from Normal$(0, 0.5)$. As above, curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method. FDP curves are shown with solid lines and corresponding FDR estimates are shown with dotted lines. Only points with FDR/FDP < 0.1 are shown. On these plots, methods whose FDR curve is *below* the FDP curve can be considered to be calibrated.

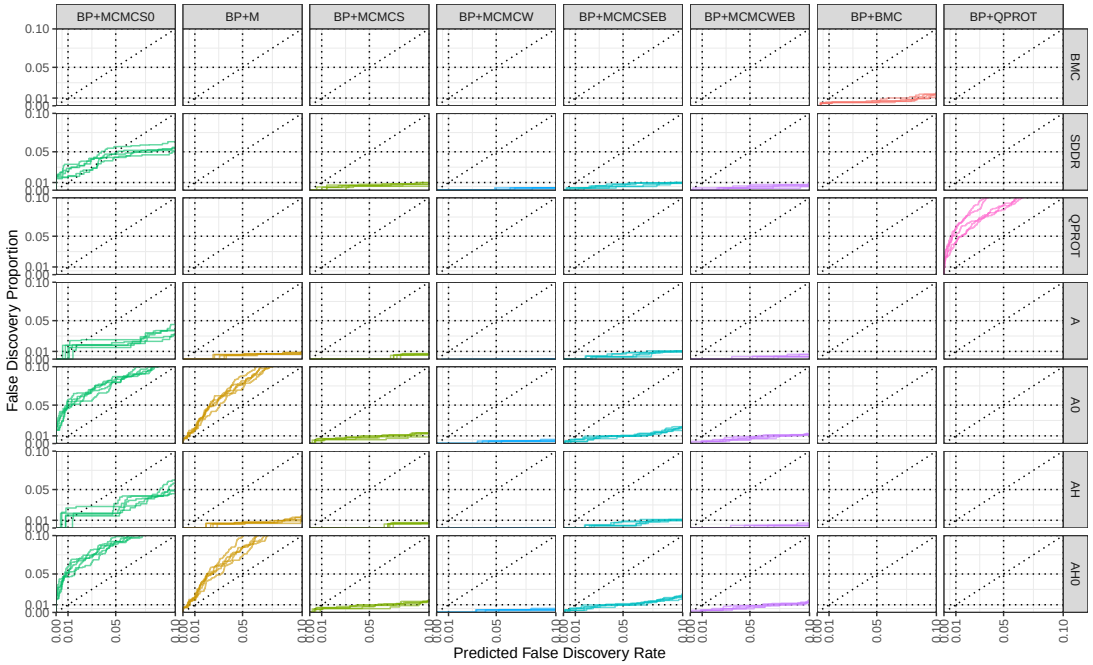

Figure 5.12: FDP vs FDR curves for the simulated data with fold-changes drawn from Normal$(0, 0.5)$. Curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method.

Table 5.6: Calibrated recall values for the simulated data with fold-changes drawn from Normal$(0, 0.5)$

	Calibrated Recall		
Method	1% FDR	5% FDR	10% FDR
BP+BMC	$\mathbf{295.2 \pm 9.63}$	$\mathbf{440.2 \pm 7.53}$	$\mathbf{527.6 \pm 11.4}$
BP+QPROT			
BP+MCMCS0+A		169 ± 16.2	282.2 ± 20
BP+MCMCS0+A0			
BP+MCMCS0+AH		182.4 ± 15.4	305.4 ± 17.3
BP+MCMCS0+AH0			
BP+MCMCS0+SDDR			386.6 ± 9.18
BP+M+A	52 ± 7.11	171.6 ± 14.5	312 ± 16.1
BP+M+A0			
BP+M+AH	56.4 ± 7.54	185.8 ± 14.7	337.8 ± 16.4
BP+M+AH0			
BP+MCMCS+A	47.4 ± 8.5	133 ± 12.2	223.4 ± 16.5
BP+MCMCS+A0	248.8 ± 9.36	377.4 ± 9.13	468 ± 13.9
BP+MCMCS+AH	48 ± 9.14	133.8 ± 12.4	224.8 ± 16.6
BP+MCMCS+AH0	249.2 ± 8.96	378 ± 9.41	467.6 ± 14.7
BP+MCMCS+SDDR	211.8 ± 6.06	304.8 ± 8.7	371.4 ± 6.88
BP+MCMCW+A			11.8 ± 10.6
BP+MCMCW+A0	131.8 ± 6.76	217.8 ± 7.4	286 ± 9.7
BP+MCMCW+AH			12.8 ± 9.86
BP+MCMCW+AH0	131.4 ± 7.06	218 ± 7.65	286.2 ± 9.42
BP+MCMCW+SDDR	123.8 ± 7.16	196.8 ± 6.1	250 ± 6.89
BP+MCMCSEB+A	119 ± 10.1	287.2 ± 11.6	413.2 ± 11.5
BP+MCMCSEB+A0	$\mathbf{323.8 \pm 11.8}$	$\mathbf{475.6 \pm 12.3}$	$\mathbf{582.2 \pm 15.5}$
BP+MCMCSEB+AH	119 ± 9.97	287.4 ± 10.8	414.2 ± 11.5
BP+MCMCSEB+AH0	$\mathbf{324.8 \pm 11.1}$	$\mathbf{476 \pm 12.3}$	$\mathbf{582.8 \pm 16.7}$
BP+MCMCSEB+SDDR	263.6 ± 7.77	371 ± 8	443 ± 7.31
BP+MCMCWEB+A	58.6 ± 10.3	179.2 ± 14.5	297 ± 13.8
BP+MCMCWEB+A0	276.8 ± 7.6	421.8 ± 7.6	525.4 ± 15.7
BP+MCMCWEB+AH	59 ± 10.1	178.6 ± 14.9	297.8 ± 13.5
BP+MCMCWEB+AH0	276.6 ± 7.09	421.6 ± 6.84	525 ± 16.5
BP+MCMCWEB+SDDR	226 ± 5.15	331.2 ± 5.72	401.6 ± 2.88

For those methods using BayesProt input data, mean recalls across five replicated BayesProt runs are shown, along with the estimated standard deviation. The values for the three best-performing methods in each column are highlighted in bold for emphasis.

5.6.6 Simulated Data — **Normal**$(0, 1.0)$

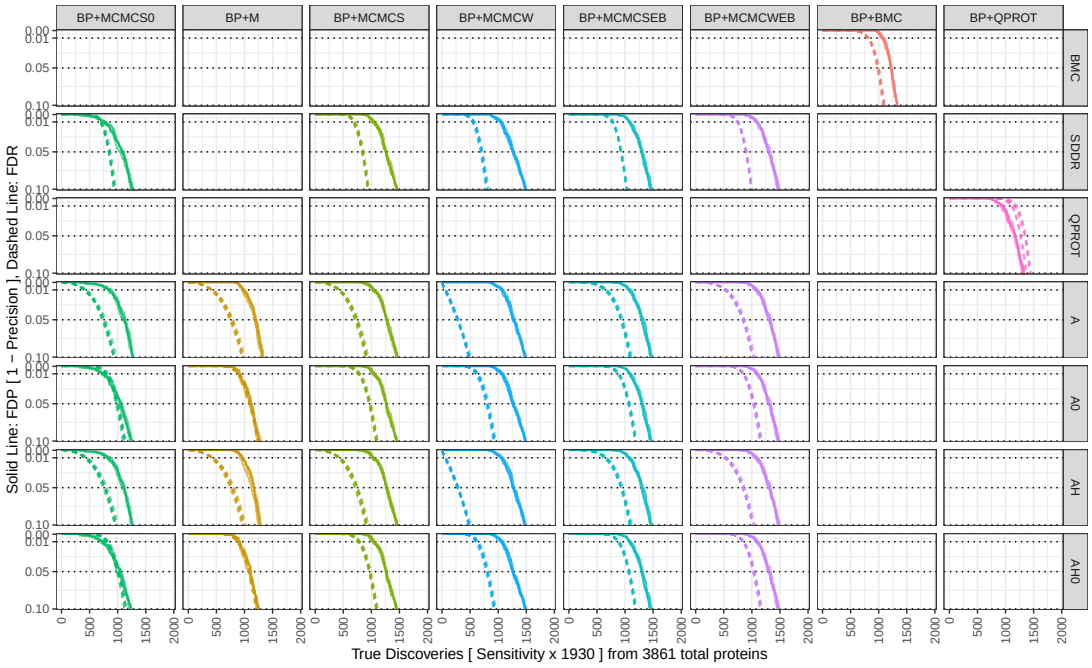

Figure 5.13: Precision–recall curves for the simulated data with fold-changes drawn from Normal$(0, 1.0)$. As above, curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method. FDP curves are shown with solid lines, and corresponding FDR estimates are shown with dotted lines. Only points with FDR/FDP < 0.1 are shown. On these plots, methods whose FDR curve is *below* the FDP curve can be considered to be calibrated.

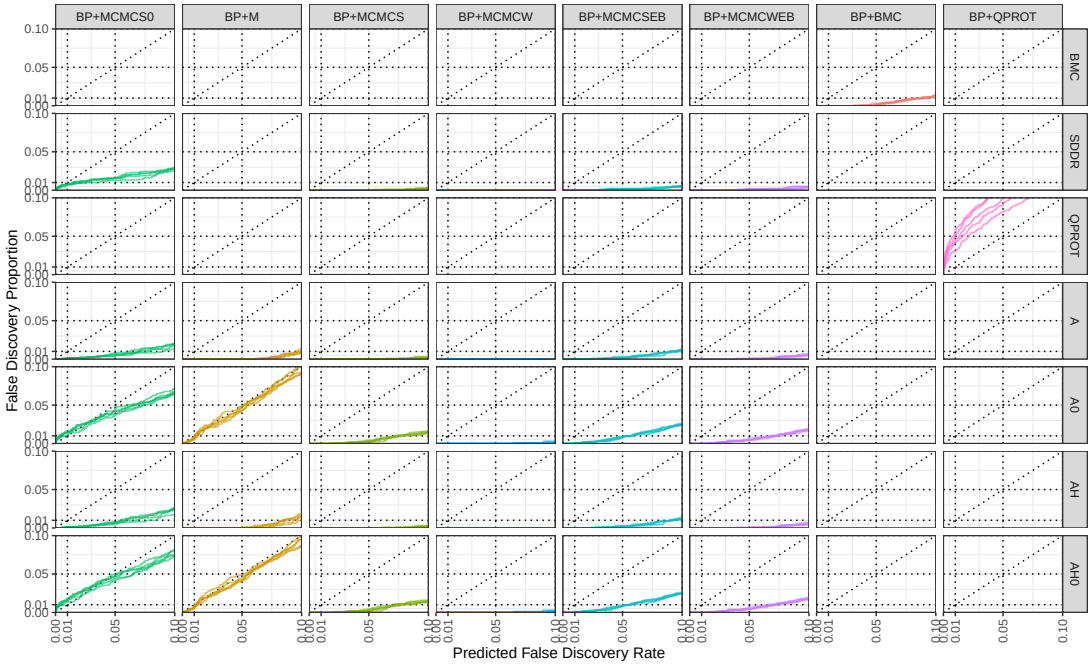

Figure 5.14: FDP vs FDR curves for the simulated data with fold-changes drawn from Normal$(0, 1.0)$. Curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method.

Table 5.7: Calibrated recall values for the simulated data with fold-changes drawn from Normal$(0, 1.0)$

	Calibrated Recall		
Method	1% FDR	5% FDR	10% FDR
BP+BMC	812.2 ± 6.46	989.4 ± 13.6	1092 ± 14
BP+QPROT			
BP+MCMCS0+A	415 ± 17.1	747.2 ± 20.2	941.2 ± 21.4
BP+MCMCS0+A0		991 ± 19.9	1119 ± 18.7
BP+MCMCS0+AH	430.4 ± 17.5	771.2 ± 23.2	971.2 ± 21.7
BP+MCMCS0+AH0		1006.2 ± 17.2	1140.2 ± 17.8
BP+MCMCS0+SDDR	701.8 ± 14.7	848.6 ± 13.5	943 ± 14.5
BP+M+A	416.4 ± 15.6	765.6 ± 23.7	972.6 ± 20
BP+M+A0	$\mathbf{893.2 \pm 15}$	$\mathbf{1099.2 \pm 14.6}$	
BP+M+AH	424.4 ± 16.9	778 ± 28	985.2 ± 26.4
BP+M+AH0	$\mathbf{885 \pm 15.1}$	$\mathbf{1085 \pm 15}$	$\mathbf{1219.6 \pm 10.1}$
BP+MCMCS+A	424.2 ± 13.9	733 ± 23	918.4 ± 20.9
BP+MCMCS+A0	781.6 ± 17.4	970.2 ± 14.8	1095.6 ± 15.6
BP+MCMCS+AH	425.2 ± 15.1	734.6 ± 23.8	920.4 ± 20.3
BP+MCMCS+AH0	781.8 ± 17.4	970.8 ± 14.1	1097.2 ± 14.3
BP+MCMCS+SDDR	690 ± 15.7	842.6 ± 13.7	944 ± 12.3
BP+MCMCW+A	57 ± 12.3	279.6 ± 11.7	484.2 ± 16.3
BP+MCMCW+A0	624.6 ± 18.7	806.2 ± 17.3	932.8 ± 16.5
BP+MCMCW+AH	57.4 ± 12.6	280.4 ± 11.4	484.6 ± 15.7
BP+MCMCW+AH0	624.6 ± 18.5	806.2 ± 17.3	932.8 ± 16.2
BP+MCMCW+SDDR	557.6 ± 15.3	701.8 ± 15.9	803.4 ± 14.3
BP+MCMCSEB+A	653.6 ± 20.8	935.8 ± 14.5	1095.6 ± 17.5
BP+MCMCSEB+A0	882.2 ± 10.3	1069.8 ± 13.2	$\mathbf{1189.6 \pm 13.4}$
BP+MCMCSEB+AH	654 ± 20.8	935.6 ± 14.5	1096 ± 17.4
BP+MCMCSEB+AH0	$\mathbf{882.4 \pm 10.1}$	$\mathbf{1070.2 \pm 13.1}$	$\mathbf{1190.8 \pm 13.8}$
BP+MCMCSEB+SDDR	778.8 ± 10.1	928.6 ± 8.44	1031 ± 8.22
BP+MCMCWEB+A	529.8 ± 18.8	846.8 ± 17.2	1024 ± 16.9
BP+MCMCWEB+A0	846 ± 10.1	1035.8 ± 14.4	1161.8 ± 16.3
BP+MCMCWEB+AH	530 ± 19	846.8 ± 17.2	1024.2 ± 17
BP+MCMCWEB+AH0	846.2 ± 9.86	1035.8 ± 14.4	1162.2 ± 15.7
BP+MCMCWEB+SDDR	745.8 ± 9.12	895.4 ± 9.4	998 ± 10

For those methods using BayesProt input data, mean recalls across five replicated BayesProt runs are shown, along with the estimated standard deviation. The values for the three best-performing methods in each column are highlighted in bold for emphasis.

5.6.7 Simulated Data — Normal$(0.5, 0.25)$

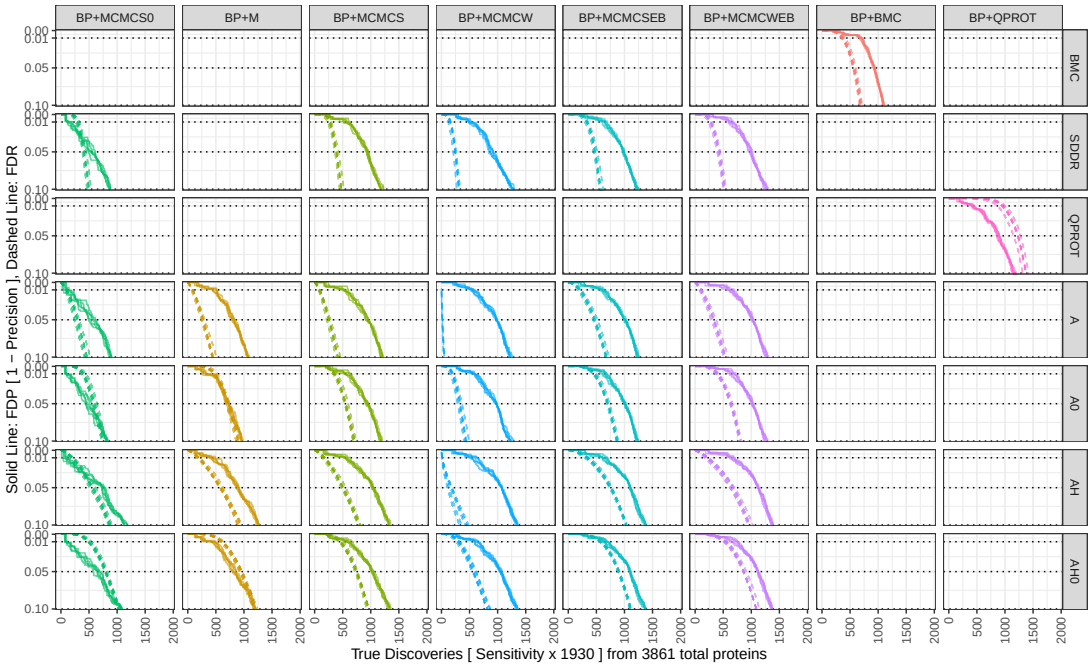

Figure 5.15: Precision–recall curves for the simulated data with fold-changes drawn from Normal$(0.5, 0.25)$. As above, curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method. FDP curves are shown with solid lines, and corresponding FDR estimates are shown with dotted lines. Only points with FDR/FDP < 0.1 are shown. On these plots, methods whose FDR curve is *below* the FDP curve can be considered to be calibrated.

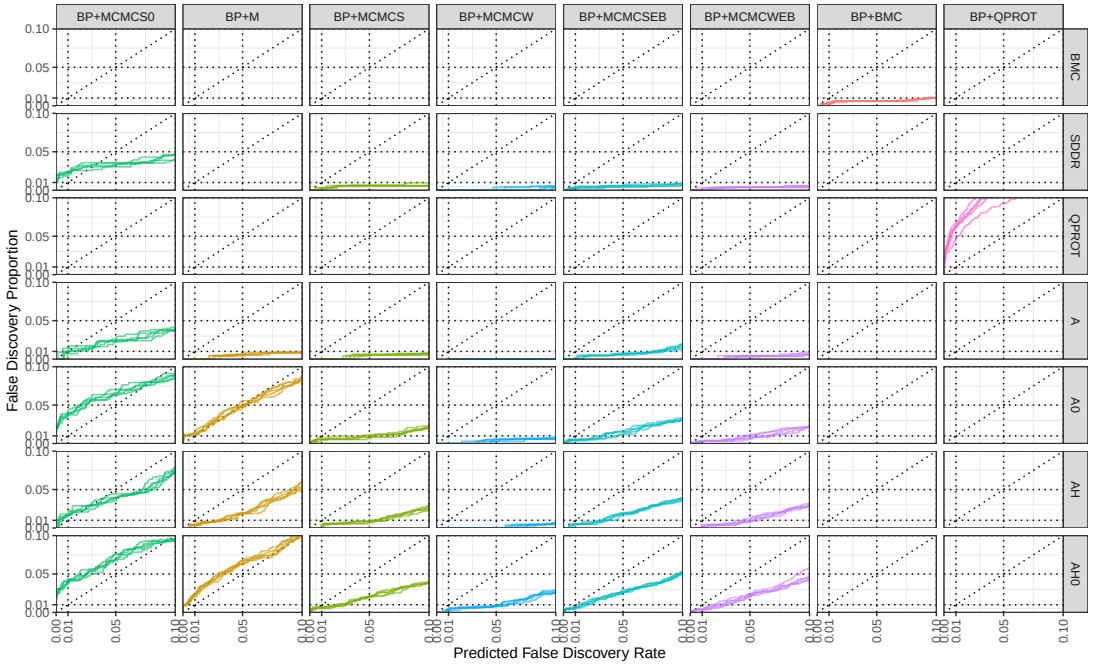

Figure 5.16: FDP vs FDR for the simulated data with fold-changes drawn from Normal$(0.5, 0.25)$ Curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method.

Table 5.8: Calibrated recall values for the simulated data with fold-changes drawn from Normal$(0.5, 0.25)$

	Calibrated Recall		
Method	1% FDR	5% FDR	10% FDR
BP+BMC	375.2 ± 22.8	571.6 ± 18.9	692.6 ± 18.8
BP+QPROT			
BP+MCMCS0+A		282 ± 32.9	460 ± 41.2
BP+MCMCS0+A0			754.8 ± 28.7
BP+MCMCS0+AH		562.4 ± 38.4	873.2 ± 23
BP+MCMCS0+AH0			$\mathbf{1031.6 \pm 18.2}$
BP+MCMCS0+SDDR		411.8 ± 24.7	496.8 ± 28.9
BP+M+A	81 ± 10.2	269.6 ± 21.6	455.4 ± 31.1
BP+M+A0			895 ± 25.7
BP+M+AH	187.6 ± 16.6	590.4 ± 24.3	924.6 ± 29.3
BP+M+AH0			
BP+MCMCS+A	82.2 ± 11.1	251.8 ± 23.7	416.4 ± 28.9
BP+MCMCS+A0	356.2 ± 23.1	557.4 ± 25.2	706 ± 19.8
BP+MCMCS+AH	170.2 ± 16.5	507.6 ± 28.1	795 ± 18.9
BP+MCMCS+AH0	$\mathbf{459.2 \pm 26}$	743.2 ± 17.1	954.8 ± 16.3
BP+MCMCS+SDDR	265.2 ± 22.6	388.6 ± 22.4	478.2 ± 28
BP+MCMCW+A		15.2 ± 3.42	48.4 ± 8.44
BP+MCMCW+A0	188.8 ± 19.5	325.6 ± 28.9	438.6 ± 34.3
BP+MCMCW+AH	12 ± 4.74	153.2 ± 34.1	406.4 ± 66.7
BP+MCMCW+AH0	299.4 ± 25.5	581.8 ± 27.7	832 ± 27
BP+MCMCW+SDDR	151.8 ± 17.7	240.8 ± 20.3	307.6 ± 23.5
BP+MCMCSEB+A	201.2 ± 20.9	474.2 ± 27.9	684.2 ± 22.8
BP+MCMCSEB+A0	461 ± 22.2	706.2 ± 17.3	870 ± 14.4
BP+MCMCSEB+AH	331.8 ± 24.2	$\mathbf{753.8 \pm 16.3}$	1030 ± 8.86
BP+MCMCSEB+AH0	$\mathbf{579.8 \pm 21.8}$	$\mathbf{897 \pm 11.2}$	$\mathbf{1107 \pm 10.4}$
BP+MCMCSEB+SDDR	327.8 ± 20.3	478.4 ± 23.8	581.2 ± 23.7
BP+MCMCWEB+A	99.4 ± 15.9	328 ± 30.7	540.8 ± 28.4
BP+MCMCWEB+A0	392.2 ± 18.6	638.2 ± 17.8	805 ± 12.5
BP+MCMCWEB+AH	210.2 ± 27.3	658.4 ± 34.2	975.2 ± 31.5
BP+MCMCWEB+AH0	$\mathbf{534.8 \pm 21.7}$	$\mathbf{872.8 \pm 22.6}$	$\mathbf{1101 \pm 25.3}$
BP+MCMCWEB+SDDR	272.4 ± 22.5	419.4 ± 20.2	524.8 ± 19.8

For those methods using BayesProt input data, mean recalls across five replicated BayesProt runs are shown, along with the estimated standard deviation. The values for the three best-performing methods in each column are highlighted in bold for emphasis.

5.6.8 Simulated Data — **Normal**$(-0.5, 0.25)$ + **Normal**$(0.5, 0.25)$

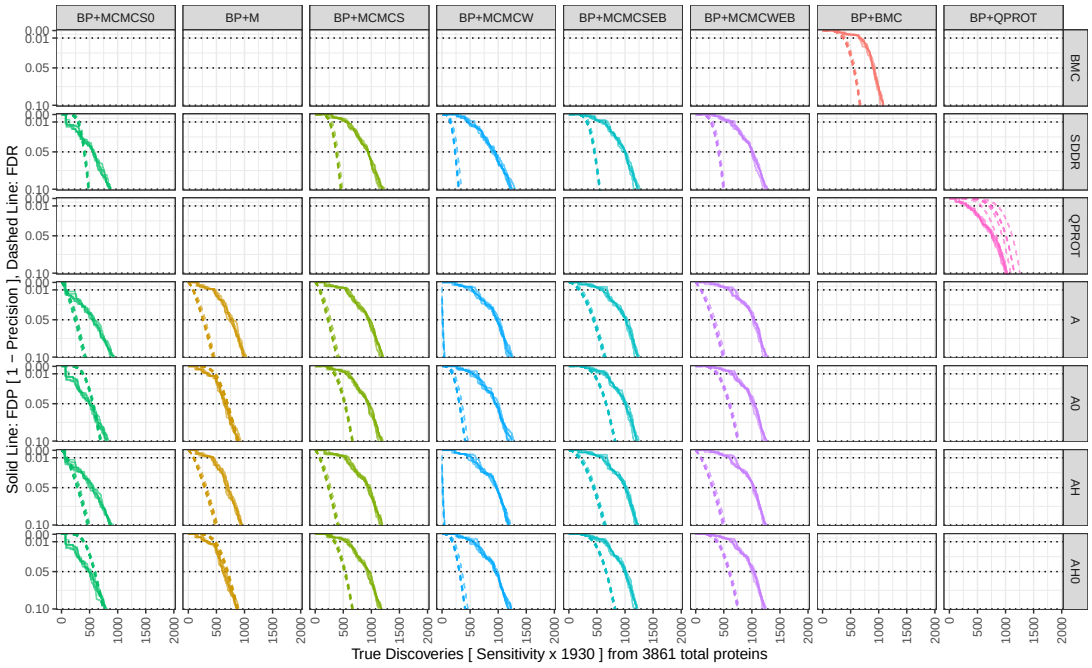

Figure 5.17: Precision–recall curves for the simulated data with fold-changes drawn from Normal$(-0.5, 0.25)$ + Normal$(0.5, 0.25)$. As above, curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method. FDP curves are shown with solid lines, and corresponding FDR estimates are shown with dotted lines. Only points with FDR/FDP < 0.1 are shown. On these plots, methods whose FDR curve is *below* the FDP curve can be considered to be calibrated.

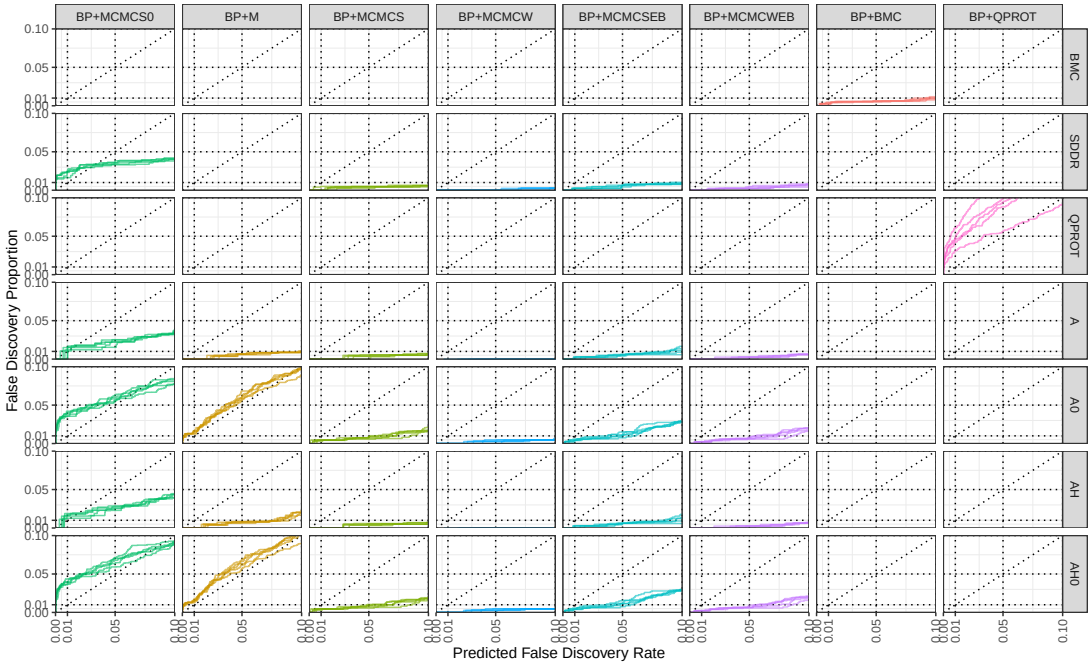

Figure 5.18: FDP vs FDR curves for the simulated data with fold-changes drawn from Normal$(-0.5, 0.25)$ + Normal$(0.5, 0.25)$. Curves are separated on input data, quantification method and FDR estimation method and coloured according to quantification method.

Table 5.9: Calibrated recall values for the simulated data with fold-changes drawn from Normal$(-0.5, 0.25)$ + Normal$(0.5, 0.25)$

	Calibrated Recall		
Method	1% FDR	5% FDR	10% FDR
BP+BMC	375.4 ± 17.6	556 ± 11.5	668.6 ± 7.89
BP+QPROT			
BP+MCMCS0+A		255.8 ± 21.8	416.8 ± 16.3
BP+MCMCS0+A0			698.4 ± 9.81
BP+MCMCS0+AH		293.4 ± 20.8	477.8 ± 19.6
BP+MCMCS0+AH0			743.4 ± 8.96
BP+MCMCS0+SDDR		405.8 ± 11.1	484.6 ± 10.5
BP+M+A	82.4 ± 5.32	269.4 ± 19	448 ± 18.6
BP+M+A0			$\mathbf{880.6 \pm 14.4}$
BP+M+AH	91.8 ± 6.1	300.6 ± 19.7	497.6 ± 20.9
BP+M+AH0			
BP+MCMCS+A	79.6 ± 8.73	240 ± 22.9	393 ± 18.5
BP+MCMCS+A0	345.6 ± 15	527.8 ± 11.1	663.8 ± 8.41
BP+MCMCS+AH	80.6 ± 7.7	245.6 ± 18.5	401 ± 14.3
BP+MCMCS+AH0	346.4 ± 15.3	528.4 ± 11.1	665.4 ± 8.2
BP+MCMCS+SDDR	265.2 ± 19.8	379.8 ± 15.4	462.6 ± 13.7
BP+MCMCW+A		10.6 ± 5.86	42.2 ± 7.4
BP+MCMCW+A0	188.6 ± 16.4	318.2 ± 21.3	419.6 ± 23.7
BP+MCMCW+AH		11 ± 6.04	42.2 ± 7.12
BP+MCMCW+AH0	188.8 ± 16.3	318.4 ± 21.4	420.2 ± 23.4
BP+MCMCW+SDDR	152.4 ± 12.7	241.2 ± 20.5	307.8 ± 22
BP+MCMCSEB+A	190.4 ± 20.2	445 ± 16.8	638.8 ± 16.2
BP+MCMCSEB+A0	$\mathbf{438.6 \pm 12.6}$	$\mathbf{663.8 \pm 8.26}$	$\mathbf{821.6 \pm 12.6}$
BP+MCMCSEB+AH	190.4 ± 20.3	445 ± 16.4	639 ± 16.3
BP+MCMCSEB+AH0	$\mathbf{438.6 \pm 12.2}$	$\mathbf{663.6 \pm 8.02}$	$\mathbf{821.4 \pm 12.4}$
BP+MCMCSEB+SDDR	323.6 ± 16.4	457.2 ± 9.65	550.8 ± 8.64
BP+MCMCWEB+A	90.4 ± 6.58	307.6 ± 20.3	500 ± 20.8
BP+MCMCWEB+A0	383.8 ± 14.8	597.4 ± 13.5	756.8 ± 12.4
BP+MCMCWEB+AH	90.6 ± 6.35	308 ± 20	500 ± 20.6
BP+MCMCWEB+AH0	$\mathbf{383.8 \pm 14.8}$	$\mathbf{598 \pm 13.4}$	757.2 ± 11.6
BP+MCMCWEB+SDDR	277.2 ± 19.7	409.4 ± 12	500.6 ± 10.5

For those methods using BayesProt input data, mean recalls across five replicated BayesProt runs are shown, along with the estimated standard deviation. The values for the three best-performing methods in each column are highlighted in bold for emphasis.

5.7 Computational Speedup

The reduction in computation time required to estimate differential expression estimates for a whole data set is apparent when the Bayesian model comparison is compared with an existing method which utilises MCMC to fit a similar model.

The runtime of the Bayesian model comparison procedure was compared with that of the QPROT[43] software, using the parameters for number of MCMC burn-in iterations and sampling iterations that were used to generate results in their original paper (10,000 burn-in and 100,000 sampling iterations).

Analysis was carried out on a Linux workstation with an Intel® Xeon® CPU (E5–/1620 v3) @ 3.50GHz, running Ubuntu 18.04 and R version 3.5.2. For the QPROT analysis version, 1.3.5 was used. Timing of both methods was recorded and logged using the `system.time` function in R.

The results of 100 repetitions of the analysis of a simulated data set of 5000 proteins in a 10 vs 10 experiment are shown in Figure 5.19. Over the 100 runs, a mean speed-up of 1020× was observed.

It should also be noted that, by the nature of MCMC, QPROT provides only an approximation of the posterior distribution of log-fold-changes for each protein. Meanwhile, the analytic approach gives exact distributions which could be used in any number of ways, including generating Z-statistics.

Furthermore, it should be noted that QPROT's results are not entirely reproducible: random number generation in the software is determined by a hard-coded random seed and the input data. In fact, changing the ordering of proteins in the input file also affects the results for a given number of MCMC iterations. In contrast the analytic Bayesian model comparison is entirely deterministic.

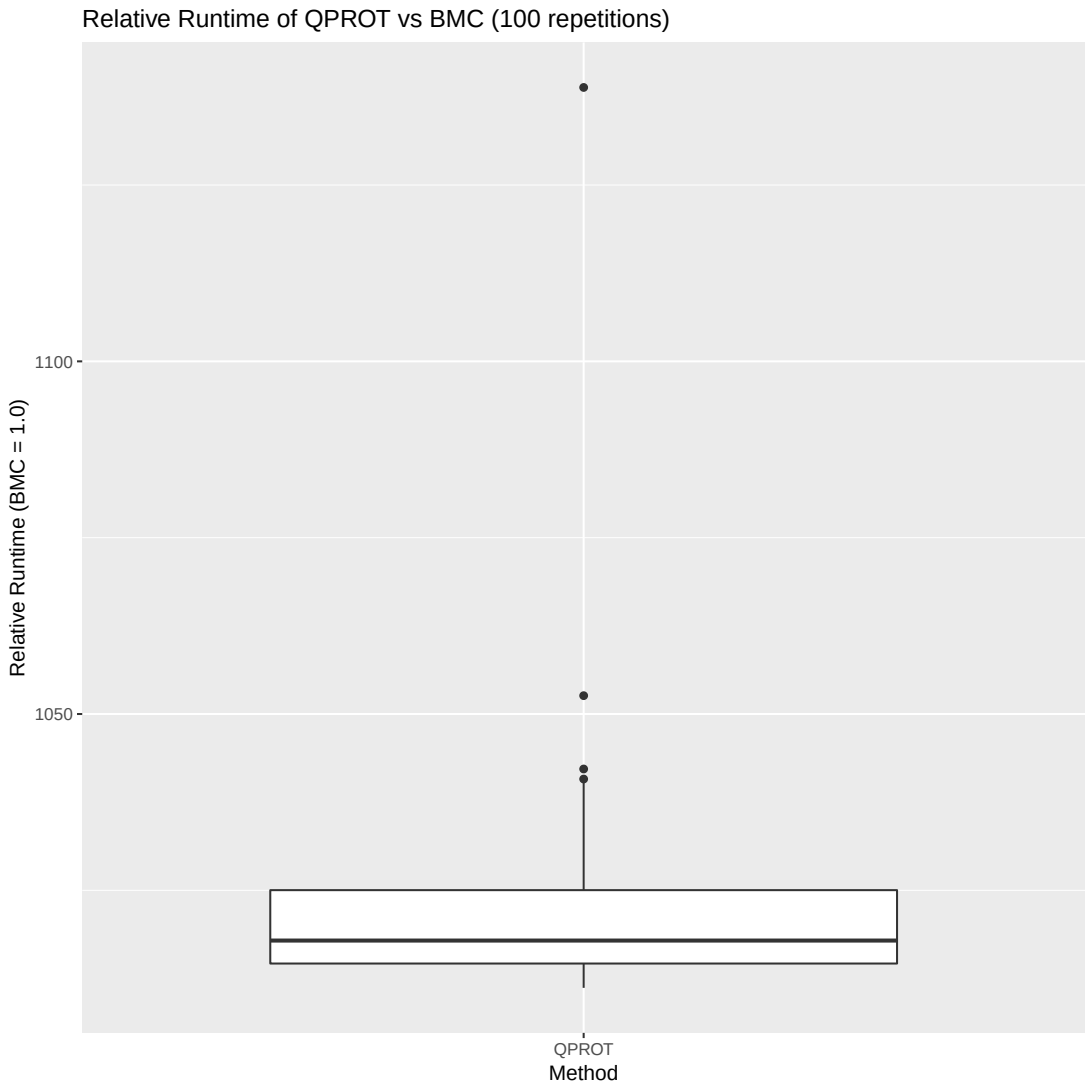

Figure 5.19: Relative runtime of QPROT software versus the Bayesian model comparison on a simulated data set of 5000 proteins in a 10 vs 10 experiment over 100 runs. The observed speedup of the Bayesian model comparison versus QPROT is over 1000.

5.8 Discussion

5.8.1 Spike-in Data

Firstly, consider the results from the spike-in data sets.

Single Fractions

As was already noted in Chapter 3, the MCMC method which ignores protein quantification uncertainty (MCMCS0) has reduced recall at higher precisions (e.g. 1% and 5%) versus the other methods, which all incorporate quantification uncertainty (see Figure 5.3). For this method the FDR is uncalibrated for all configurations of ash (see Figure 5.4).

The metafor t-test achieves much lower recall at high precisions in combination with the ash methods and with inconsistent FDR calibration (see Table 5.2 Figure 5.4).

For the MCMC-based quantification methods, the addition of the limma-derived empirical Bayes priors on residual variance has very little effect on the observed recall, though it results in FDRs that are less conservative than without, regardless of the configuration of ash chosen. This effect is more pronounced for the unequal variances t-test (for example compare the curves for BP+MCMCW with those for BP+MCMCWEB in Figure 5.3).

QPROT's FDR estimates are significantly anti-conservative, and as such, QPROT does not show calibrated FDR at any point (see Figure 5.3).

The Bayesian model comparison gives calibrated FDR estimates across the range shown, as evidenced in Figure 5.4, the curves being below the ideal diagonal. However, as demonstrated in Table 5.2, the Bayesian model comparison shows some reduction in recall over the MCMC-based methods; this is unsurprising since the relationship between protein quantifications and their respective uncertainties is lost by the approximate method made by the Bayesian model comparison approach. Contrasting the calibrated recall in Table 5.2 of the analytic Bayesian model comparison (BMC) with the approximate model comparison afforded by the Savage–Dickey density ratio (SDDR), it can be seen that the analytic model comparison achieves higher calibrated recall than the approximate version, regardless of the underlying quantification model. The method most directly comparable to the analytic model comparison is the equal variances MCMC-based test with limma-derived empirical Bayes prior on residual variance (BP+MCMCSEB+SDDR).

Pooled Fractions

The MCMC-based methods which take protein quantification uncertainty into account (MCMCW and MCMCS) achieve greater recall but with very conservative FDR estimates when combined with the ash methods (see Figure 5.5). The MCMC method with the empirical Bayes prior on variance achieves similar recall but with much better FDR calibration.

On this more difficult data, the Bayesian model comparison maintains calibration of False Discovery rate. While it fails to achieve the highest calibrated recall at all three tested FDRs, it is arguably more consistent; for instance, Table 5.3 demonstrates that BP+MCMCSEB+AH0 achieves the highest recall at 1% FDR but fails to control the FDR at higher thresholds. Conversely, BP+MCMCWEB+AH0 achieves the highest recall at 10% FDR but with much lower recall at 1% FDR. As above, the approximate model comparison with SDDR shows much lower recall for all of the quantification models than its analytic counterpart.

Faulty Data

The results of each of the methods for differential expression and FDR estimation on this "faulty" data serve to demonstrate how each of them behaves in the presence of poor-quality data.

In a number of scenarios, ash's FDR estimates are so conservative that ash fails to identify any differentially expressed proteins at 1% or 5% reported FDR; these can be identified in Figure 5.7 where the dashed FDR curves begin below the dotted lines corresponding to 1% and 5% FDR, (e.g. BP+MCMCWEB+A).

The approximate model comparison with SDDR shows much lower recall than the analytic Bayesian model comparison: compare the calibrated recall values in Table 5.4 for the analytic model comparison (BMC) and the approximations (SDDR).

Conclusions from Real Data

It is clear from looking at the precision–recall curves in Figures 5.3, 5.5 and 5.7, and comparing the calibrated recall values for MCMCSEB versus MCMCS and MCMCWEB versus MCMCW in Tables 5.2, 5.3 and 5.4, that the use of empirical Bayes to inform priors of residual variance across proteins allows for an increase in calibrated recall.

The Bayesian model comparison technique is able to achieve consistent calibration of quantitative false discovery rates, although most often this is at the expense of lower

recall over some of the other methods. Comparing it with the MCMC-based methods which use the same empirical Bayes prior on residual variance, the Bayesian model comparison is more consistent in its calibrated recall; in some cases the MCMC-based method which performs best at, say, 1% FDR does not show calibrated FDR at 5% or 10% (for example, BP+MCMCSEB+AH0 in Table 5.3). Conversely, those methods which perform best at 10% FDR perform more poorly at 1% FDR. This is summarised in Table 5.10, showing the mean rank of the methods at each FDR threshold.

The Bayesian model comparison has relatively consistent performance across these data sets and across FDR thresholds compared to many of the other methods tested above, including the approximate model comparison with the Savage–Dickey density ratio. It should be noted that ash's modelling of the distribution of fold-changes across the population puts it at a significant advantage for these data sets, since the true distribution of fold-changes is asymmetric.

Table 5.10: Mean ranking of methods across the three spike-in data sets

Method	Mean Rank 1% FDR	5% FDR	10% FDR
BP+BMC	6.20	6.67	10.00
BP+QPROT	18.67	19.87	21.13
BP+MCMCS0+A	15.67	14.27	14.67
BP+MCMCS0+A0	18.67	19.87	19.13
BP+MCMCS0+AH	17.93	15.93	16.47
BP+MCMCS0+AH0	18.67	19.87	14.80
BP+MCMCS0+SDDR	18.67	18.87	12.73
BP+M+A	14.00	10.00	8.73
BP+M+A0	15.07	13.80	15.33
BP+M+AH	4.60	15.47	16.13
BP+M+AH0	14.73	13.47	13.80
BP+MCMCS+A	17.20	16.47	16.47
BP+MCMCS+A0	7.27	7.40	9.87
BP+MCMCS+AH	11.80	5.67	10.53
BP+MCMCS+AH0	2.33	13.00	14.53
BP+MCMCS+SDDR	10.60	14.07	15.93
BP+MCMCW+A	18.67	19.80	21.13
BP+MCMCW+A0	15.67	17.13	18.80
BP+MCMCW+AH	18.67	19.13	18.53
BP+MCMCW+AH0	14.20	10.47	15.13
BP+MCMCW+SDDR	17.47	18.47	20.33
BP+MCMCSEB+A	12.73	12.20	12.13
BP+MCMCSEB+A0	5.33	5.73	7.47
BP+MCMCSEB+AH	6.60	16.27	15.67
BP+MCMCSEB+AH0	8.93	14.20	14.07
BP+MCMCSEB+SDDR	11.00	12.87	14.53
BP+MCMCWEB+A	17.33	16.47	16.33
BP+MCMCWEB+A0	9.93	10.67	11.33
BP+MCMCWEB+AH	14.27	9.93	3.73
BP+MCMCWEB+AH0	6.53	3.93	4.93
BP+MCMCWEB+SDDR	15.60	16.07	17.13

Methods are ranked by calibrated recall for each run, data set and FDR threshold before the mean rank for each FDR threshold is calculated for each method across all runs and data sets. Where methods do not give a calibrated recall at a particular FDR threshold they are considered to be in joint last place for that data set.

5.8.2 Simulated Data

The results from the simulated data sets serve to illustrate the differences exhibited by the tested methods when analysing data with more realistic distributions of fold-changes across the population of proteins than would normally be available by analysing spike-in data sets.

The data set with the smaller simulated fold-changes sampled from Normal$(0, 0.25)$ is the most "difficult" data set considered. Figure 5.9 and Table 5.5 show that the Bayesian model comparison achieves the highest calibrated recall at 1%, 5% and 10% FDR. Most of the other tested methods exhibit very conservative FDR estimates and therefore comparatively low calibrated recall at all three thresholds.

Figures 5.11 and 5.13 and Tables 5.6 and 5.7 demonstrate that with the simulated fold-changes from Normal$(0, 0.5)$, and Normal$(0, 1.0)$ the Bayesian model comparison is bested by other methods. Only with the relatively large fold-changes sampled from Normal$(0, 1.0)$ does the metafor t-test show results comparable to the MCMC-based methods (see Table 5.7).

The simulated data with biased fold-changes (Normal$(0.5, 0.25)$) demonstrates the advantage of the half-uniform component distribution in ash, which achieves a significant gain in calibrated recall over its counterparts with uniform component distribution; as can be seen, for example, by comparing results in Table 5.8 for BP+MCMCSEB+AH0 with results for BP+MCMCSEB+A0. Here the Bayesian model comparison falls behind; its lack of awareness of the population-level distribution of fold-changes puts it at a disadvantage behind the MCMC-based methods combined with ash's modelling of fold-change distributions.

In the case where the distribution of effects sizes are not unimodal (Normal$(\pm 0.5, 0.25)$), that is, contrary to the unimodal assumption made by ash, it might be expected that the performance of ash would be affected negatively. However Figure 5.17 and Table 5.9 show that, consistent with the findings in [61], ash appears to be robust against the violation of this unimodal assumption. With a symmetric distribution of fold-changes, the Bayesian model comparison is at less of a disadvantage but still falls behind.

As above, the mean ranking of each method at each FDR threshold across the simulated data sets is presented in Table 5.11.

As with the spike-in data sets, while not achieving the highest overall recall the Bayesian model comparison shows consistent calibration of FDR across the five simulated data sets.

Table 5.11: Mean ranking of methods across the five simulated data sets.

	Mean Rank		
Method	1% FDR	5% FDR	10% FDR
BP+BMC	4.40	5.64	8.36
BP+QPROT	23.88	27.00	26.52
BP+MCMCS0+A	20.96	21.68	21.08
BP+MCMCS0+A0	23.88	22.48	16.32
BP+MCMCS0+AH	21.32	18.08	15.80
BP+MCMCS0+AH0	23.88	22.20	13.84
BP+MCMCS0+SDDR	20.48	16.64	15.52
BP+M+A	20.32	20.84	19.76
BP+M+A0	18.28	17.36	13.68
BP+M+AH	17.92	16.92	15.20
BP+M+AH0	18.64	21.20	20.56
BP+MCMCS+A	20.84	23.64	25.24
BP+MCMCS+A0	7.24	8.52	9.96
BP+MCMCS+AH	20.04	21.08	22.08
BP+MCMCS+AH0	6.48	6.60	7.96
BP+MCMCS+SDDR	10.12	13.84	17.48
BP+MCMCW+A	23.36	26.12	27.56
BP+MCMCW+A0	13.44	16.68	22.04
BP+MCMCW+AH	23.28	25.72	27.12
BP+MCMCW+AH0	12.96	14.88	19.36
BP+MCMCW+SDDR	14.88	20.76	23.80
BP+MCMCSEB+A	13.44	12.36	12.56
BP+MCMCSEB+A0	2.28	2.84	4.20
BP+MCMCSEB+AH	12.68	10.04	9.68
BP+MCMCSEB+AH0	2.04	2.12	2.28
BP+MCMCSEB+SDDR	9.36	9.72	11.72
BP+MCMCWEB+A	18.56	18.52	17.88
BP+MCMCWEB+A0	5.12	5.60	6.88
BP+MCMCWEB+AH	17.76	16.40	15.44
BP+MCMCWEB+AH0	4.92	4.76	5.00
BP+MCMCWEB+SDDR	9.92	12.36	14.84

Methods are ranked by calibrated recall for each run, data set and FDR threshold before the mean rank for each FDR threshold is calculated for each method across all runs and data sets. Where methods do not give a calibrated recall at a particular FDR threshold they are considered to be in joint last place for that data set.

5.8.3 General Observations

As was discussed in Chapter 3, there is a slight advantage gained by those differential testing methods that utilise the uncertainty in the inputted protein quantification estimates.

When the false discovery rate reported by these methods is compared with the actual false discovery proportion, the FDR is particularly conservative, until a more informative prior on variance is introduced, highlighting the importance of regularisation of variances across the population. The effect of a more informative prior on residual variance is more readily observable in the case of an unequal variances Bayesian t-test: the unequal variances t-test necessitates that the same data previously used for estimating one variance parameter is now required for estimating two.

The population-level modelling of the distribution of fold-changes provided by ash has a clear advantage in data sets where the true distribution of fold-changes is asymmetric. Such distributions are common in spike-in benchmarking data sets. The ash R package is capable of providing calibrated FDR, though which configuration is most suitable appears to be dependent on the quantification method chosen and the desired FDR cutoff.

Conversely, despite its use of a population-based model of fold-change, QPROT consistently underestimates the false discovery rate.

The Bayesian model comparison is a Bayesian method producing posterior error probabilities which do not require any correction: hence the FDR estimates follow as a natural consequence of the mean probability across the significant set. Furthermore, as shown in Section 5.7, the Bayesian Model Comparison procedure can be performed with much less computation than many of the other methods compared in this chapter which rely on MCMC sampling for Bayesian inference. It has been demonstrated that the method's performance in terms of calibrated recall is consistent across a range of FDR thresholds. It should also be noted that the method used to account for protein quantification uncertainty is an approximation; the relationship between individual protein quantification estimates and their uncertainties is lost by the algorithm. This is in contrast to the manner in which quantification uncertainty is handled by the MCMC-based testing. The empirical Bayes priors on variance can also be thought of as approximations to a hierarchical model with hyper-priors on the variance parameters across proteins. Hence, a number of possible avenues for future work can be identified, which will discussed in the Section 5.9.1.

5.9 Conclusions

This chapter has demonstrated that statistical testing based on Bayesian model comparison has the potential to give more accurate FDR estimates than a number of other methods. As well as simultaneously calculating estimates for differential expression and hypothesis testing, the analytic approach adopted does not require costly numerical integration, thus leading to a method for differential expression analysis that is computationally inexpensive and reproducible.

There is some potential for improvement to this work, which would serve as avenues for future research.

5.9.1 Future Work

Outlier Rejection and Unequal Variances

The use of Student-t distributions in place of normal distributions would offer some robustness to outliers[122], though this version of the model would not have an analytic posterior, and as such would require numerical integration (e.g. MCMC) to fit the models.

Another development that could be made would be to expand the model comparison methodology to assigning separate variances to each sample group. This would correspond to the Welch's t-test used in frequentist statistics and to the unequal variances Bayesian t-test described in Section 3.3. The above results demonstrate that some small gains in recall can be made in the transition between equal and unequal variances, although effort should be made to ensure that FDR control is not lost. Again, this model would no longer have an analytic posterior, requiring MCMC or similar to be applied.

While both of these modifications would mean that the Bayes Factor was no longer available analytically, a number of techniques exist in the literature for estimating the value of the Bayes factor from MCMC samples (see Section 2.3.9 in Chapter 2), other than the Savage–Dickey density ratio which was evaluated above in Section 5.5. These methods may produce better estimates of the Bayes factor for this application and should be evaluated.

Inference of Null Proportion

In Section 5.2.1, a simple assumption was made: the model prior ratio, $\frac{P(M_N)}{P(M_F)}$, the relative probability of the two models was set to 1. This could instead be used as an

additional parameter which imparts prior knowledge regarding the expected proportion of null vs non-null proteins in the data set. This is conceptually similar to the π_0 null proportion parameter which other methods for FDR estimation, such as ash and Storey's qvalue, make efforts to estimate.

In the case of the Bayesian model comparison, altering this parameter does not affect the ordering of candidates in the outputted results, only rescaling the posterior error probabilities and hence the predicted FDR. The mapping from Bayes factor to posterior error probability is illustrated for a number of different values of the model prior ratio in Figure 5.20.

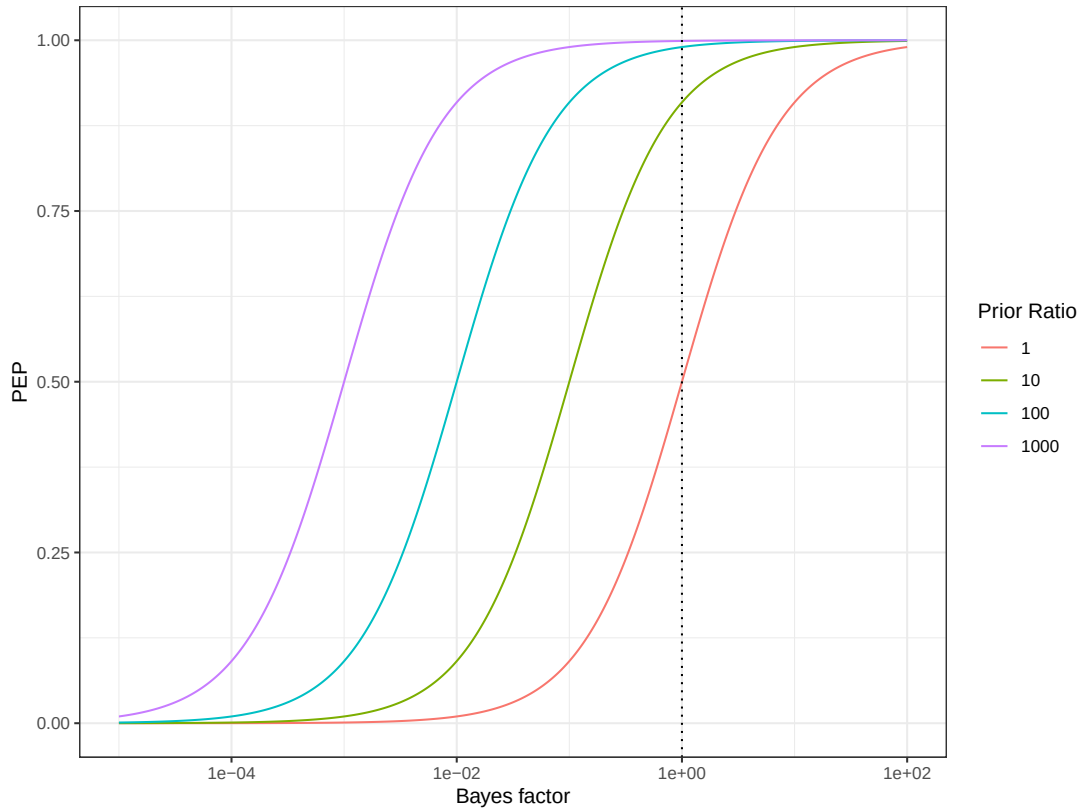

Figure 5.20: Mapping of Bayes factor to posterior error probability for a number of different values of the model prior ratio, $\frac{P(M_N)}{P(M_F)}$. Note that Bayes factor is shown on a log-10 scale. Proteins believed to be differentially expressed have low values for the Bayes factor and proteins believed to be non-differentially expressed have high values for the Bayes factor. Increasing the model prior ratio gives more conservative PEP values but does not change the ordering of proteins.

The estimation of π_0 forms a central part of a number of the methods for FDR estimation examined in this chapter and in Chapter 3, and so it would be appropriate that future developments on this work incorporate similar ideas.

All Proteins at Once

A number[48][38] of models for quantitative proteomics analysis described in earlier work have advocated that models should analyse the entire population of proteins at once; indeed in [39], Gelman et al. suggested that such a hierarchical model removes the need for multiple hypothesis correction. Fitting models to all proteins in parallel would allow for the fitting of hyperparameters, which allows for the borrowing of strength between proteins in order to better control the false discovery rate and better estimate π_0.

Population Distribution of Fold-Changes

For many of the data sets tested, the population-level modelling of fold-changes employed by the ash R package achieves greater recall of differentially expressed proteins when compared with the Bayesian model comparison testing.

The analysis of all proteins at once would facilitate the use of a hyper-prior on fold-change, thus allowing for modelling of the distribution of fold-changes across the population of proteins thereby allowing for the σ_d parameter to be calibrated for each data set.

Alternatively, an empirical Bayes approach similar to that used in ash might be employed to estimate this hyperparameter.

5.9.2 Final Remarks

In light of the above results, it can be concluded that while the modelling of fold-change distributions across the population of proteins does have some advantages over assumed independence, the calibration of false discovery rate is by no means guaranteed. It has been shown that the Bayesian model comparison technique can achieve consistent calibration of FDR. Hence, it seems likely that the fusion of these techniques could yield a best-of-both-worlds solution, giving improved recall while at the same time giving properly calibrated FDR and this would therefore be an ideal target for future work in this area.

Chapter 6

A Shared Peptide Model for Proteoform-level Analysis

6.1 Introduction

The term "proteoform" is used to describe all of the proteins which arise as the product of a single gene[46]. These proteins can differ slightly in their amino acid sequence, either due to genetic variation (different alleles of the same gene), differences in RNA transcription or post-translational modifications. These proteins are similar in their amino acid composition, resulting in sub-strings of amino acids that are identical between the proteoforms. Upon enzymatic digestion, these sub-strings of the original proteoforms, the shared peptides in question, are now indistinguishable from each other when observed by a mass spectrometer. Furthermore, shared peptides can also arise through random chance; with a limited pool of possible amino acids, short strings of amino acids can result in shared peptides between otherwise unrelated proteins. It has been estimated that shared peptides account for as much as 50% of the data in a protein database[47].

As discussed in Chapter 2, shared peptides present a challenging problem. Their inclusion in analysis requires the proportions of the contributions from their parent proteoforms to be modelled.

This chapter proposes a Bayesian hierarchical model which explicitly models the underlying proteoform-level abundances, allowing for quantification of proteoforms. Results from a number of small data sets are presented, comparing the results with those obtained by an equivalent model which does not account for shared peptides.

Section 6.2 describes the proposed model, detailing how proteoform-level abun-

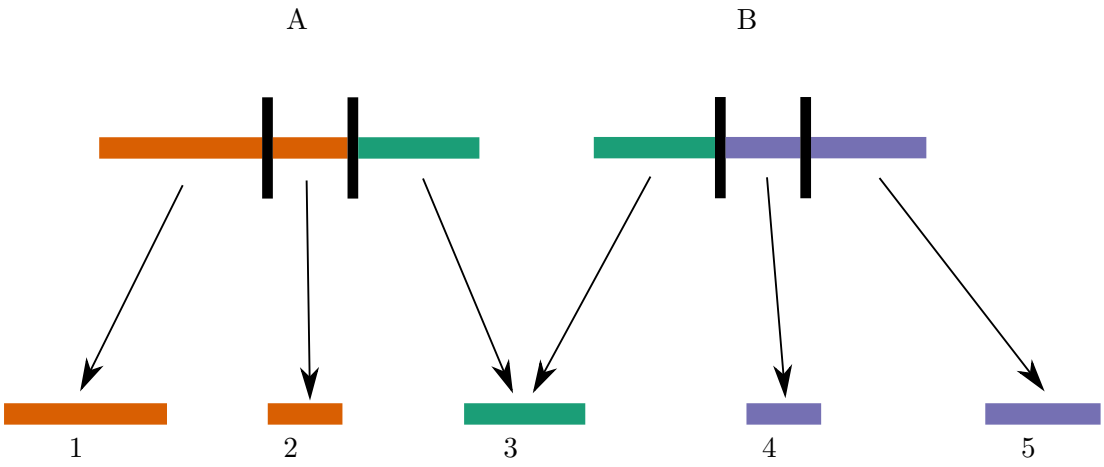

Figure 6.1: Illustration of shared and unique peptides being produced by enzymatic digestion of two proteoforms. The two proteoforms A and B are similar along some sub-string of their sequence. Upon digestion, proteoform A produces peptides 1, 2 and 3. Proteoform B produces peptides 3, 4 and 5. The resulting observed ion counts of the shared peptide 3 cannot be determined as having been produced by either proteoform A or B.

dances are inferred from peptide-level abundances, which are themselves inferred from feature-level intensities.

Section 6.5 presents results from simulated data, demonstrating the model's utility and scenarios in which the model can and cannot make inferences on the relative quantifications between proteoforms, that is, absolute quantification relative to a baseline proteoform. A possible application is highlighted wherein the relative abundance of unrelated proteins might be inferred through an artificial "bridge" protein.

In Section 6.6 results from spike-in data are presented, including an example similar to a hypothetical bridge protein.

Section 6.7 applies the model to a small part of a clinical data set, showing the potential to infer relative quantification of a proteoform whose sequence is a subset of another proteoform.

Section 6.8 presents a high-level discussion of the results in this chapter.

Finally, Section 6.9 ends the chapter with some concluding remarks, highlighting some potential areas for future research.

6.2 Model for Shared Peptide Quantification

A generative, hierarchical model is presented, which attempts to model the processes by which the data is generated, from proteoform-level abundances to peptide abundances to the resulting ion counts. Firstly, the process by which assay-level proteoform abundances are generated is described.

6.2.1 Proteoform-level Abundances

The simplest effects to model are those which happen at the proteoform level. Here, it suffices to consider a single proteoform i for the sake of simplicity.

At the top of the hierarchical model is the reference proteoform abundance of proteoform i, P_i^{Ref}. This is the estimated proteoform abundance in the assay that is arbitrarily selected to be used as the baseline. This is assigned a log-normal prior:

$$P_i^{\text{Ref}} \sim \text{Log-Normal}(a, 3) \tag{6.1}$$

where a is estimated from the log of the median of the observed counts, thus ensuring that the prior distribution is reasonable for the data set being considered (see Section 6.3.3).

Next the systematic deviations of the other assays relative to the reference assay are considered; $\beta_{i,k}^{\text{Assay}}$ is assigned a normal prior:

$$\beta_{i,k}^{\text{Assay}} \sim \text{Normal}(0, 3) \tag{6.2}$$

Then the assay-level abundance of proteoform i in assay k can be written:

$$P_{i,k} = P_i^{\text{Ref}} \cdot \exp(\beta_{i,k}^{\text{Assay}}) \tag{6.3}$$

6.2.2 Proteoform-Peptide Relationships

Next, peptide-level abundances are calculated from proteoform level abundances according to the relationship between peptides and their parent proteoform or proteoforms. Take for example, two proteoforms A and B with unique peptides 1 and 3 with a shared peptide 2, depicted in Figure 6.15. If the abundance of proteoform A in sample

k is denoted as $P_{A,k}$ and the abundance of proteoform B in sample k as $P_{B,k}$, then the expected abundances of peptides 1, 2 and 3 ($Q_{1,k}$, $Q_{2,k}$ and $Q_{3,k}$) can be calculated:

$$Q_{1,k} = P_{A,k} \tag{6.4}$$
$$Q_{2,k} = P_{A,k} + P_{B,k} \tag{6.5}$$
$$Q_{3,k} = P_{B,k} \tag{6.6}$$

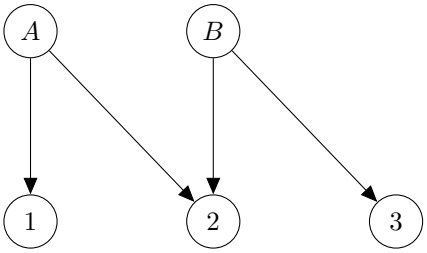

Figure 6.2: Example of two proteoforms with one shared peptide.

This relationship can be represented with a matrix multiplication:

$$\begin{bmatrix} Q_{1,k} \\ Q_{2,k} \\ Q_{3,k} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} P_{A,k} \\ P_{B,k} \end{bmatrix} \tag{6.7}$$

This can be generalised to N peptides and M proteoforms with an arbitrary relationship between them, described by an $N \times M$ matrix R:

$$\mathbf{Q_k} = \mathbf{R} \cdot \mathbf{P_k} \tag{6.8}$$

where $R_{j,i} = 1$ if peptide j is produced by proteoform i, otherwise it takes the value 0. $Q_{j,k}$ is the expected abundance of peptide j in sample k.

6.2.3 Peptide-level Effects

The effects on peptide-level abundances in each digested sample can now be applied to the underlying assay-level peptide abundances, namely the sample random effect with per-peptide variance:

$$\epsilon_{d,j}^{\text{Sample}} \sim \text{Normal}(0, \sigma_j^{\text{Peptide}}) \tag{6.9}$$

so that the peptide-level abundance for peptide j in sample s, assay k is:

$$Q_{j,k} \cdot \exp(\epsilon_{s,j}^{\text{Sample}}) \tag{6.10}$$

6.2.4 Feature Ionisation

It is expected that each peptide will be split via LC-MS into several features. This splitting into features is an imperfect process however, and some molecules will remain un-ionised or be undetected for whatever reason. To model the ionisation proportion of each feature, we can consider this missing proportion as an additional missing feature for each peptide, such that all these feature effects across a peptide in an LC-MS run sum to one.

Let the number of features (including the missing feature) for a peptide j equal F_j. Then peptide j has associated with it F_j "ionisation proportions", $\beta_{f,j}^{\text{Feature}}$ where $f = 1 : F_j$. Since these ionisation proportions must sum to one across a peptide in a given LC-MS run, together they form a point on a $(F_j - 1)$-simplex. Arbitrarily, the missing proportion is assigned the index $f = 1$ for each peptide. A Dirichlet prior is then assigned to the ionisation proportions for each peptide:

$$\boldsymbol{\beta}_{\mathbf{j}}^{\text{Feature}} \sim \text{Dirichlet}(\boldsymbol{\alpha}_{\mathbf{j}}) \tag{6.11}$$

where α_j is a F_j-element prior concentration vector for peptide j. Since the features which have been detected must have a non-zero ionisation proportion (else they would not have been detected), it is desirable to choose this prior such that minimal probability mass is placed in the region where $\boldsymbol{\beta}_{\mathbf{j}}^{\text{Feature}} \approx \begin{bmatrix} 1.0 & 0.0 & \cdots & 0.0 \end{bmatrix}$, (see Section 6.3.3) This is achieved by setting each α_j as follows:

$$\alpha_j = \begin{bmatrix} 0.1 & 1.5 & \cdots & 1.5 \end{bmatrix} \tag{6.12}$$

In reality, no peptide will have been ionised perfectly. However, it suffices to consider the relative ionisation of each peptide and feature. The abundance of a feature f

belonging to peptide j in assay k, sample s is:

$$Q_{j,k} \cdot \beta_{f,j}^{\text{Feature}} \cdot \exp(\epsilon_{s,j}^{\text{Sample}}) \tag{6.13}$$

6.2.5 Measurement-level effect

Next, there are the effects which only apply at the level of individual measurements of each feature, namely the residual error with per-feature variance.

$$\epsilon_{k,f}^{\text{Residual}} \sim \text{Normal}(0, \sigma_f^{\text{Feature}}) \tag{6.14}$$

so that observation of feature f from peptide j in assay k, sample s has the mean rate:

$$Q_{j,k} \cdot \beta_{f,j}^{\text{Feature}} \cdot \exp(\epsilon_{s,j}^{\text{Sample}} + \epsilon_{k,f}^{\text{Residual}}) \tag{6.15}$$

6.2.6 Error Model

As in the BayesProt model described in Chapter 3, a left-censored Poisson likelihood can be employed.

$$\lambda_{j,k,s,f} = Q_{j,k} \cdot \beta_{f,j}^{\text{Feature}} \cdot \exp(\epsilon_{s,j}^{\text{Sample}} + \epsilon_{k,f}^{\text{Residual}}) \tag{6.16}$$

$$y_{j,k,s,f} \sim \text{Poisson}(\lambda_{j,k,s,f}) \quad \text{for } y_{j,k,s,f} > 0 \tag{6.17}$$

$$P(y_{j,k,s,f} < C_f) = \sum_{x=0}^{C_f - 1} [\text{Poisson}(x|\lambda_{j,k,s,f})] \quad \text{for } y_{j,k,s,f} = 0 \tag{6.18}$$

where C_f is the minimum non-missing count for feature f. A similar log-Normal likelihood was also evaluated (see Section 6.4).

6.3 The Whole Model

As a whole, the model looks like this:

$$P_i^{\text{Ref}} \sim \log\text{-Normal}(a, 3) \tag{6.19}$$

$$\beta_{i,k}^{\text{Assay}} \sim \text{Normal}(0, 3) \tag{6.20}$$

$$\beta_{f,j}^{\text{Feature}} \sim \text{Dirichlet}(\alpha_j) \tag{6.21}$$

$$\epsilon_{s,j}^{\text{Sample}} \sim \text{Normal}(0, \sigma_j^{Peptide}) \tag{6.22}$$

$$\epsilon_{n,f}^{\text{Residual}} \sim \text{Normal}(0, \sigma_f^{Feature}) \tag{6.23}$$

$$P_{i,k} = P_i^{\text{Ref}} \cdot \exp(\beta_{i,k}^{\text{Assay}}) \tag{6.24}$$

$$\mathbf{Q_k} = \mathbf{R} \cdot \mathbf{P_k} \tag{6.25}$$

$$\lambda_{j,k,s,f} = Q_{j,k} \cdot \beta_{f,j}^{\text{Feature}} \cdot \exp(\epsilon_{s,j}^{\text{Sample}} + \epsilon_{k,f}^{\text{Residual}}) \tag{6.26}$$

$$y_{j,k,s,f} \sim \text{Poisson}(\lambda_{j,k,s,f}) \quad \text{for } y_{j,k,s,f} > 0 \tag{6.27}$$

$$P(y_{j,k,s,f} < C_f) = \sum_{x=0}^{C_f - 1} [\text{Poisson}(x|\lambda_{j,k,s,f})] \quad \text{for } y_{j,k,s,f} = 0 \tag{6.28}$$

The result is a model that is conceptually similar to that described in [48] but with the important difference that the model described here models the ionisation of each feature separately as "fixed effects".

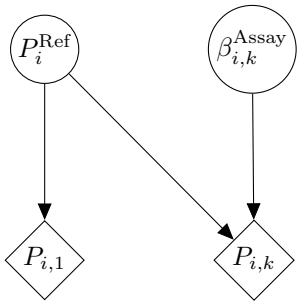

Figure 6.3: Bayesian network representation of equation (6.24).

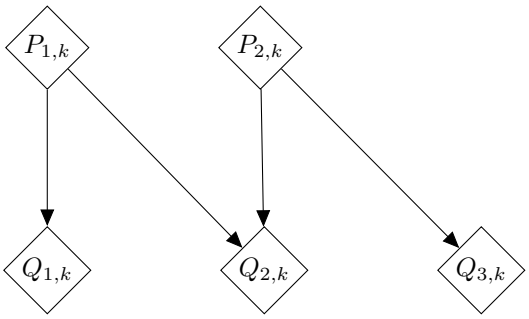

Figure 6.4: Bayesian network representation of equation (6.25) for two proteoforms P_1 and P_2, each with one unique peptide (Q_1 and Q_3 respectively) and one shared peptide (Q_2).

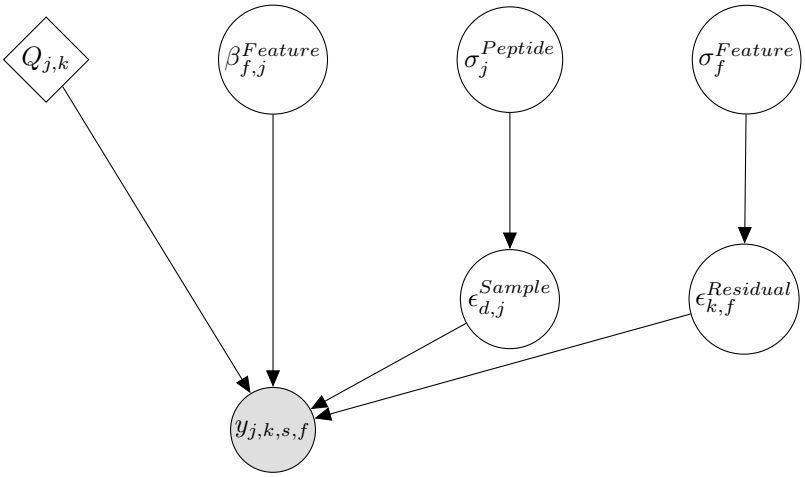

Figure 6.5: Bayesian network representation of equations (6.21), (6.22), (6.23), (6.26), (6.27) and (6.28).

6.3.1 Relation to BayesProt model

For a single proteoform, this shared peptide model is equivalent to the model considered in Chapter 3. The model for BayesProt can be expressed as:

$$y_{j,k,s,f} \sim \text{Poisson}(\exp(\beta_{f,j}^{\text{Feature-BayesProt}} + \beta_{1,k}^{\text{Assay}} + \epsilon_{s,j}^{\text{Sample}} + \epsilon_{k,f}^{\text{Residual}})) \tag{6.29}$$

for observation $y_{j,k,s,f}$ of a feature f of peptide j in assay k, sample s.

For a single proteoform, $i = 1$ at all times. Then from (6.24) above:

$$\begin{aligned} Q_{j,k} &= P_{1,k} \quad \forall j && (6.30)\\ &= P_1^{\text{Ref}} \cdot \exp(\beta_{1,k}^{\text{Assay}}) && (6.31) \end{aligned}$$

Substituting this into (6.26) and (6.27):

$$\begin{aligned} & y_{j,k,s,f} \sim \text{Poisson}(Q_{j,k} \cdot \beta_{f,j}^{\text{Feature}} \cdot \exp(\epsilon_{s,j}^{\text{Sample}} + \epsilon_{k,f}^{\text{Residual}})) && (6.32)\\ \implies & y_{j,k,s,f} \sim \text{Poisson}(P_1^{\text{Ref}} \cdot \beta_{f,j}^{\text{Feature}} \cdot \exp(\beta_{1,k}^{\text{Assay}}) \cdot \exp(\epsilon_{s,j}^{\text{Sample}} + \epsilon_{k,f}^{\text{Residual}})) && (6.33)\\ \implies & y_{j,k,s,f} \sim \text{Poisson}(P_1^{\text{Ref}} \cdot \beta_{f,j}^{\text{Feature}} \cdot \exp(\beta_{1,k}^{\text{Assay}} + \epsilon_{s,j}^{\text{Sample}} + \epsilon_{k,f}^{\text{Residual}})) && (6.34) \end{aligned}$$

Features in both models are equivalent so:

$$P_1^{\text{Ref}} \cdot \beta_{f,j}^{\text{Feature}} = \exp(\beta_{f,j}^{\text{Feature-BayesProt}}) \tag{6.35}$$

Substituting this into (6.34):

$$\begin{aligned} & y_{j,k,s,f} \sim \text{Poisson}(P_1^{\text{Ref}} \cdot \beta_{f,j}^{\text{Feature}} \cdot \exp(\beta_{1,k}^{\text{Assay}} + \epsilon_{s,j}^{\text{Sample}} + \epsilon_{k,f}^{\text{Residual}})) && (6.36)\\ \implies & y_{j,k,s,f} \sim \text{Poisson}(\exp(\beta_{f,j}^{\text{Feature-BayesProt}}) \cdot \exp(\beta_{1,k}^{\text{Assay}} + \epsilon_{s,j}^{\text{Sample}} + \epsilon_{k,f}^{\text{Residual}})) && (6.37)\\ \implies & y_{j,k,s,f} \sim \text{Poisson}(\exp(\beta_{f,j}^{\text{Feature-BayesProt}} + \beta_{1,k}^{\text{Assay}} + \epsilon_{s,j}^{\text{Sample}} + \epsilon_{k,f}^{\text{Residual}})) && (6.38) \end{aligned}$$

which is the model for BayesProt, as in Chapter 3.

Degrees of Freedom

The two models can also shown to be equivalent in terms of the number of degrees of freedom. Looking at (6.35): if there are J peptides, $j = 1 : J$, each with F_j features, including the missing features (i.e. $F_j - 1$ features are *observed* for peptide j), then by the nature of the Dirichlet distribution:

$$\sum_{f=1}^{F_j} \beta_{f,j}^{\text{Feature}} = 1 \quad \forall j \in J. \tag{6.39}$$

So for each peptide, $\beta_{f=1:F_j,j}^{\text{Feature}}$ has $F_j - 1$ degrees of freedom (which equals the number of *observed* features). Since the feature ionisations are all relative to each other, another degree of freedom is removed, meaning the total degrees of freedom for all β^{Feature} is $(\sum_{j=1}^{J} F_j - 1) - 1$, which is equal to the total number of observed features minus one. Add to this the degree of freedom offered by P_1^{Ref}, then the total degrees of freedom for $P_1^{\text{Ref}} \cdot \beta_{f,j}^{\text{Feature}}$ is equal to the number of *observed* features. Since the features being considered in the shared peptide model and the BayesProt model are equivalent, the degrees of freedom is equal to $\sum_{j=1}^{J}(F_j - 1) = \text{Num. Observed Features}$ for both sides of (6.35).

6.3.2 Relative Abundance

As a consequence of the estimation of feature weights and hence ionisation effects, we gain the additional ability to infer posterior distributions for the relative abundance across proteoforms sharing peptides. This value can easily be calculated from the MCMC samples. For two proteoforms 1 and 2 in assay k, the relative abundance between them, a ratio $P_{(2,1),k}^{\text{Rel}}$, can be calculated as:

$$P_{(2,1),k}^{\text{Rel}} = \frac{P_2^{\text{Ref}} \cdot \exp(\beta_{2,k}^{\text{Assay}})}{P_1^{\text{Ref}} \cdot \exp(\beta_{1,k}^{\text{Assay}})} \tag{6.40}$$

In effect, per-assay absolute quantification of proteoforms can be achieved relative to a baseline proteoform.

6.3.3 Stan Model

The model described above is implemented in the Stan[81] programming language. In order to make the Stan model file as generic as possible, due to some limitations of the Stan modelling language and for the purposes of numerical stability, the model is not implemented exactly as described above.

Dirichlet Prior on β^{Feature}

Since the β^{Feature} parameter needs to be normalised for each peptide it would be most convenient to set the data type for these parameters to be of the `simplex` type in Stan, which is a specialised version of a vector which is automatically constrained to be normalised (with looser checking of the constraints to account for numerical inaccuracy near the boundaries).

However, each peptide can have a differing number of features, which would require an array of these simplexes of differing sizes (known as a ragged array). Stan currently does not permit ragged arrays.

One approach would be to implement these Dirichlet-distributed parameters as a single vector parameter θ with prior $\Gamma(\alpha_j, 1.0)$, where subsections of the whole vector are normalised appropriately:

$$\theta \sim \Gamma(\alpha_j, 1.0) \tag{6.41}$$

$$\beta_j^{\text{Feature}} = \theta_j / \sum_{i=1}^{K} \theta_i \tag{6.42}$$

This is equivalent to applying a Dirichlet prior. However, this model, while "correct" is non-identifiable, due to the extra degree of freedom introduced by having K gamma-distributed parameters, compared with the $K-1$ degrees of freedom in a Dirichlet-distribution.

Instead, we initialise $K-1$ parameters, $\theta_{j=1:K-1}$ which are then transformed to K parameters, $\beta_{1:K}^{\text{Feature}}$:

$$\beta_j^{\text{Feature}} = \frac{\theta_j}{(1 + \sum_{i=1}^{K-1} \theta_i)} \quad \text{for } j = 1 : K-1 \tag{6.43}$$

$$\beta_{j=K}^{\text{Feature}} = \frac{1}{(1 + \sum_{i=1}^{K-1} \theta_i)} \tag{6.44}$$

such that the determinant of the Jacobian of this transformation for each sub-vector is:

$$\left|J\right| = \left(1 + \sum_{j=1}^{K-1} \theta_j\right)^{-K} \tag{6.45}$$

the derivation of which is shown in Appendix C.

A Dirichlet prior is then applied to the transformed parameters. Since the Dirichlet distribution is applied to the transformed parameters rather than the parameters, a Jacobian adjustment is required to account for the reparameterisation, adding the log-determinant of the Jacobian of the transform at each iteration to the log-posterior.

Protein and Peptide-level Abundances

Further to this, rather than directly working with protein and peptide abundances, we instead work with log-transformed abundances in order maintain numerical stability. This also requires that the matrix multiplication in equation (6.8) is replaced with:

$$\log\left(\mathbf{Q_k}\right) = \log\left(\mathbf{R} \cdot \exp\left(\log\left(\mathbf{P_k}\right)\right)\right) \tag{6.46}$$

However, were this to be implemented verbatim, numerical stability would not be maintained due to the moving between log and real scales. Instead, this matrix multiplication is replaced by a loop across the non-zero elements of $\mathbf{R}$ which utilises Stan's `logsumexp` function to accumulate the peptide log-abundances, $\log\left(\mathbf{Q_k}\right)$, thereby ensuring numerical stability.

Non-centred Parameterisation

Furthermore, β^{Sample}, ϵ^{Sample} and $\epsilon^{\text{Residual}}$ are implemented as having standard normal priors which are appropriately rescaled by their standard deviation before being used in calculations or output. This approach is recommended in [123] for hierarchical models in order to increase the efficiency of the HMC sampler.

Additionally, the ϵ^{Sample} parameters are constrained such that they sum to zero across each peptide. The same constraint is applied as standard for random effects in MCMCglmm and is adopted here for consistency.

Initialisation of the NUTS Sampler and Prior Parameters

The high-dimensionality of the parameter space when the model is applied to real data (such as that analysed in Section 6.7) results in a highly concentrated posterior. With wide priors and using the default initialisation of the NUTS sampler, for some data sets it was found that MCMC chains would not reliably converge to the typical set, resulting in one or two chains exploring a different region of sample space from the rest.

The priors on the feature weights were chosen so that the sampler avoided the region of parameter space where $\beta_{f,j}^{\text{Feature}} \approx 0$ for the non-missing features. Similarly, the priors on the reference level abundance and assay effects were chosen to restrict the prior from "impossible" regions of parameter space (e.g. where the abundance of a proteoform is near-zero). Additionally the Stan sampling was slightly constrained, using a smaller initial step size of 0.01 (where the default is 1.0) and a reduced initial radius for starting unconstrained parameter values (1.0 instead of the default 2.0). This resulted in better mixing of the MCMC chains.

6.4 Comparison of Poisson and Log-normal Likelihoods

The Poisson likelihood described in Section 6.2 can be replaced with a log-normal likelihood.

$$\theta_{j,k,s,f} = Q_{j,k} \cdot \beta_{f,j}^{\text{Feature}} \cdot \exp(\epsilon_{s,j}^{\text{Sample}}) \tag{6.47}$$

$$y_{j,k,s,f} \sim \text{log-Normal}(\theta_{j,k,s,f}, \sigma_f^{Residual}) \quad \text{for } y_{j,k,s,f} > 0 \tag{6.48}$$

$$P(y_{j,k,s,f,n} < C_f) = \int_{x=0}^{C_f - 1} [\text{log-Normal}(x|\theta_{j,k,s,f}, \sigma_f^{Residual})] \quad \text{for } y_{j,k,s,f} = 0 \tag{6.49}$$

Note that the residual error is instead incorporated into the log-normal likelihood, rather than being considered as a separate random effect as it is in the model with a Poisson likelihood.

The effect of applying Poisson and log-normal likelihoods to the shared peptide model was investigated by comparing the error in estimating both the within and between proteoform fold-changes to simulated data sets.

The data sets were simulated with the Julia programming language, using Julia's pseudo-random number generation and the Distributions.jl package to simulate Poisson noise of ion counts. Two proteoforms, A and B, each with one unique peptide and sharing a single peptide were simulated. Each peptide had three features simulated,

with ionisation coefficients randomly sampled from Dirichlet distributions. The same ionisation coefficients were used to simulate each data set.

Two groups of four assays were simulated. The $\log_2$-intensity of proteoform A in the first group was varied between 3.0 and 9.0 at intervals of 0.5. and the $\log_2$-fold-change of proteoform A between the two groups was varied between -2 and 2 at intervals of 0.5 (excluding 0.0). The $\log_2$-intensity of proteoform B was kept constant at 6.0 across both groups of assays.

A data set was simulated for each combination of $\log_2$-intensity and $\log_2$-fold-change of proteoform A. The shared peptide model was fitted to each simulated data set, once with a Poisson likelihood and once with a log-normal likelihood. Each was run for 5000 iterations (of which the first 2500 were discarded as warm-up) on each of four MCMC chains.

The RMS errors in estimating the $\log_2$-fold-change of proteoform A and in estimating the $\log_2$-ratio between proteoforms A and B were calculated for each of the two models for each simulated data set. The differences in RMS error between the two models was then calculated for each simulated data set.

The differences in RMS error in estimating the $\log_2$ fold change of proteoform A are presented in Figure 6.6. Similarly, the difference in RMS error of the estimated $\log_2$-ratio between the two proteoforms is presented in Figure 6.7.

In general, the shared peptide model with the log-normal likelihood exhibited lower RMS error than the model with the Poisson likelihood. The Poisson likelihood model only exhibited lower error in cases where the $\log_2$ intensity of proteoform A was extremely low. For this reason the shared peptide model with log-normal likelihood was adopted for the rest of the analysis in this chapter.

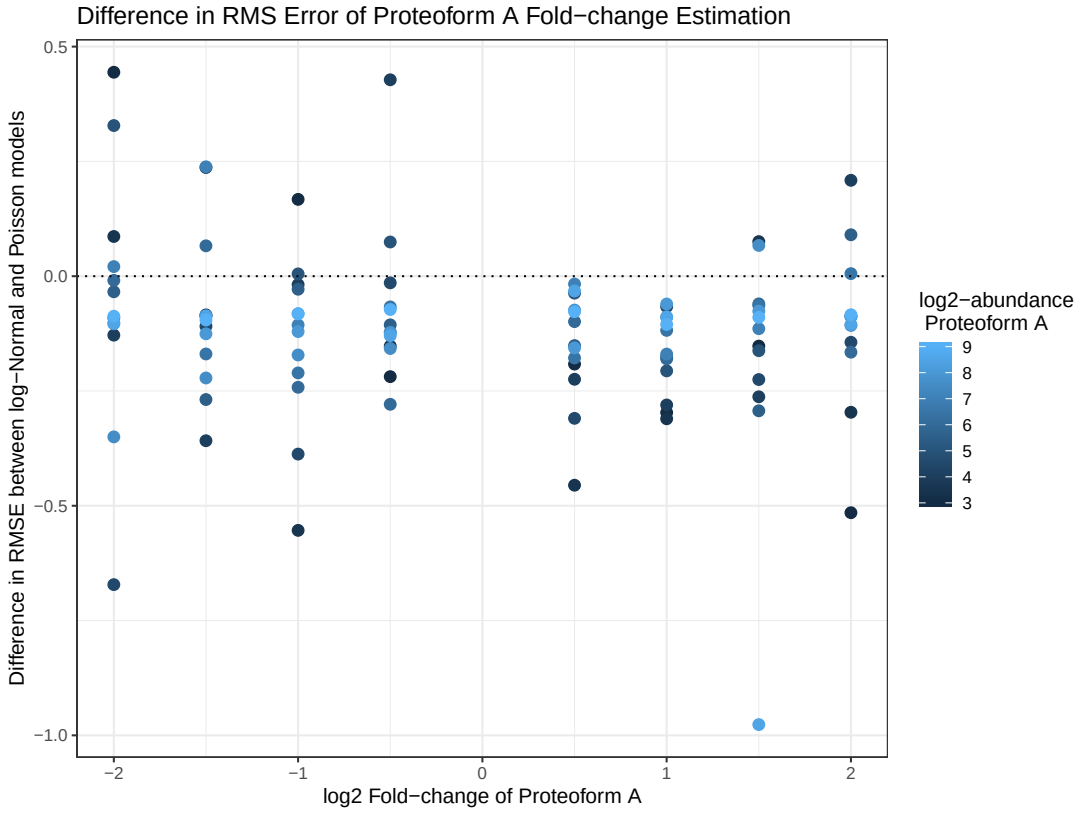

Figure 6.6: Differences in RMS error in estimating $\log_2$-fold-change of proteoform A between models with Poisson and log-normal likelihoods. Dotted line at $y = 0$ corresponds to equal RMS error for the Poisson and log-normal likelihood models. Above this line the Poisson model has lower RMS error and below, the log-normal model has lower RMS error. Points are coloured according to the $\log_2$-intensity of proteoform A in the first group of simulated assays, such that darker tones correspond to lower intensities.

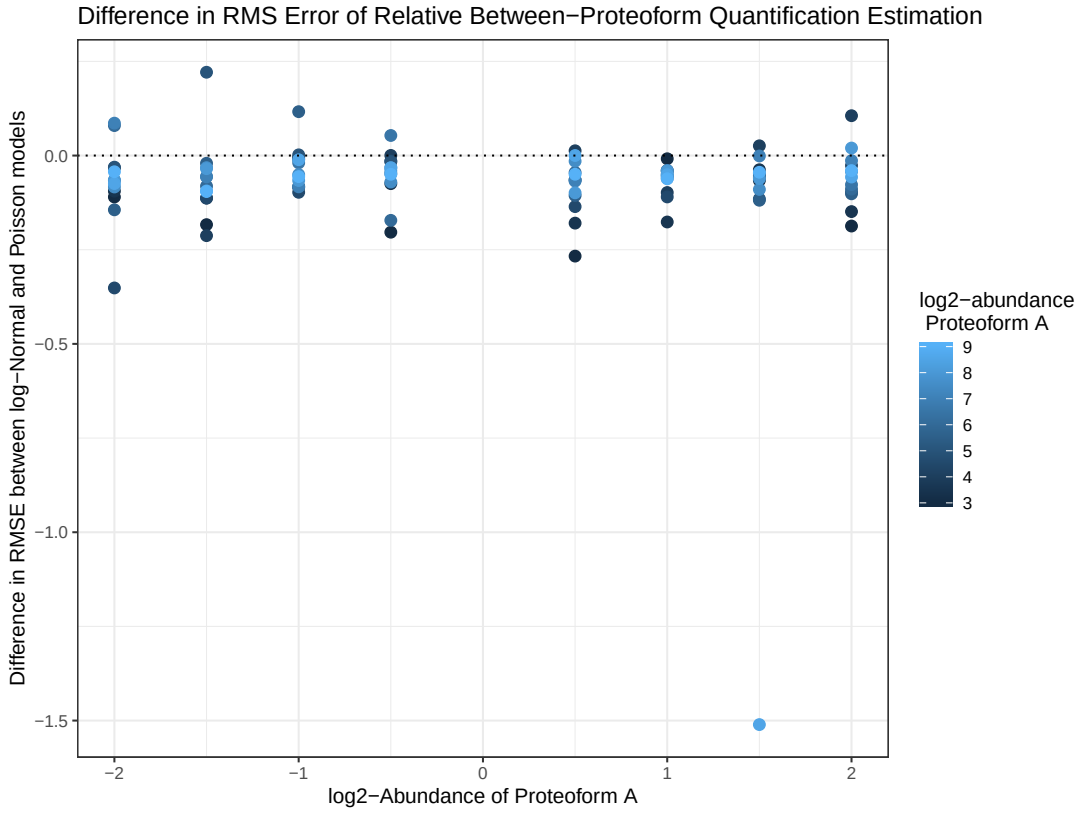

Figure 6.7: Differences in RMS error in estimating $\log_2$-ratio of proteoform A and B between models with Poisson and log-normal likelihoods. Dotted line at $y = 0$ corresponds to equal RMS error for the Poisson and log-normal likelihood models. Above this line the Poisson model has lower RMS error and below, the log-normal model has lower RMS error. Points are coloured according to the $\log_2$-intensity of proteoform A in the first group of simulated assays, such that darker tones correspond to lower intensities.

6.5 Results on Simulated Shared Peptide Data

We here present results from simulated data.

6.5.1 Two Proteoforms Sharing a Single Peptide

The simplest case is that of two proteoforms with one peptide shared between them, as in the example in Figure 6.15.

Each peptide had three features simulated with their ionisation coefficients sampled from Dirichlet distributions. Sample variance and feature variance were set to zero. While this low-noise data is unrealistic as a representation of real data it allows us to observe the limitations of the modelling of shared peptides. Finally, Poisson noise is applied to generate the simulated counts. The simulated data was varied along four axes: the baseline $\log_2$-intensity of proteoform A, A_{Ctrl}; the simulated $\log_2$-fold-change of proteoform A, $A_{\log_2\text{-Fold-Change}}$; the baseline $\log_2$-intensity of proteoform B, B_{Ctrl}; and the simulated $\log_2$-fold-change of proteoform B, $B_{\log_2\text{-Fold-Change}}$ The values used were:

$$A_{\text{Ctrl}} = 10, 11, 12 \tag{6.50}$$

$$A_{\log_2\text{-Fold-Change}} = -1, 0, 1 \tag{6.51}$$

$$B_{\text{Ctrl}} = 10, 11, 12 \tag{6.52}$$

$$B_{\log_2\text{-Fold-Change}} = -1, 0, 1 \tag{6.53}$$

simulating all possible combinations of the above to generate 81 sets of data which were then analysed with the shared peptide model described above. For each data set, four MCMC chains were run for 5000 iterations (the first 2500 being discarded as warm-up).

The RMS errors in estimating the $\log_2$-fold-change of proteoform A were calculated. Figure 6.8 shows a heatmap of the RMS error for varying $\log_2$-intensity and $\log_2$-fold-change of both proteoforms.

The error in estimating the $\log_2$-fold-change of proteoform A depends not only on the true $\log_2$-intensity and $\log_2$-fold-change of proteoform A but also on the $\log_2$-intensity and $\log_2$-fold-change of proteoform B. The signal of the shared peptide contributed by B has an effect; where this signal contributed by B is low (where B is of lower intensity than A), the error in estimating the $\log_2$-fold-change of A is lower. This is to be expected, since in this scenario, the signal of the shared peptide is more representative of the underlying signal of A. The converse also applies; where the intensity of B is higher than that of A, the shared peptide's signal is more representative of the

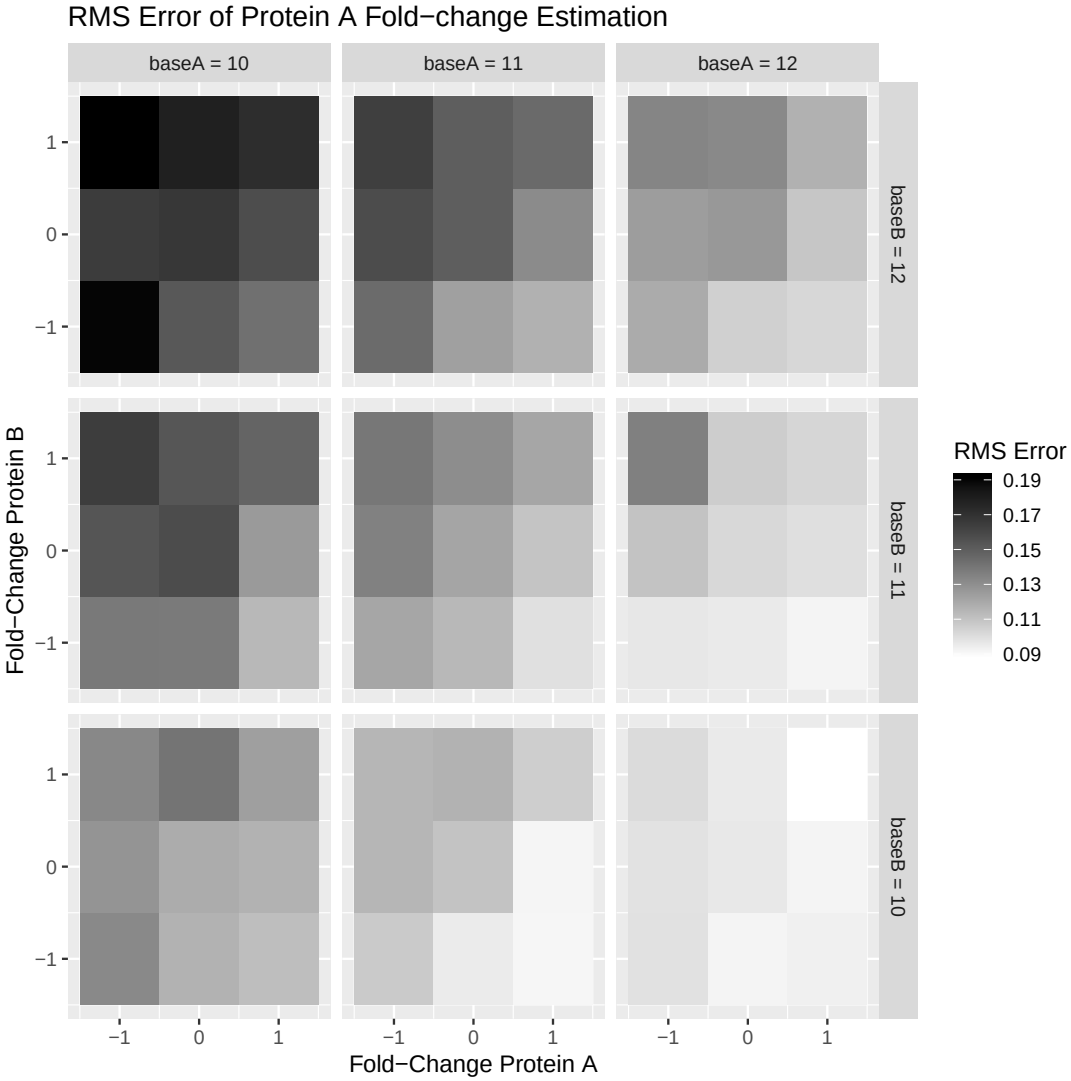

Figure 6.8: Heatmaps showing the RMS error in estimation of within-proteoform quantification for proteoform A for the case described in Section 6.5.1 with one shared peptide. Larger error is shown in darker tones. Sub-plots are grouped by the baseline intensity of proteoforms A and B on the x and y axes respectively. The x and y axes of each of the subplots correspond to the log-2 fold-change between the two sample groups of proteoforms A and B respectively.

underlying signal of B. It should be noted that this is a worst-case scenario, with both proteoforms having only one unique peptide each.

The error in estimating the relative quantification between the two proteoforms can also be calculated. Figure 6.9 shows a heatmap of the RMS error in estimating the relative quantification between proteoforms A and B across all eight simulated assays.

The main conclusion that should be drawn from Figure 6.9 is that the confidence of relative quantification between the proteoforms is dependent on the magnitude of that difference. Where the relative quantification between the two proteoforms is large, there is greater error in the estimation of this quantity. Where the proportion of the signal contributed to the shared peptide from one proteoform is "overshadowed" by the signal from the other proteoform, the model is less able to discern this proportion accurately and hence the error in determining the relative abundances increases.

On this simulated data, the model displays some bias in determining the between-proteoform relative quantifications. Indeed, in the cases where the two proteoforms have the same differential expression pattern (e.g. where both A and B have a $\log_2$-fold-change of 0.0), the model should be unable to infer the relative abundance of the two proteins. Closer inspection of the posterior samples suggest that the model is non-identifiable for these simulated data sets with only a few peptides; similar issues were not present when the model was applied to real data (see Sections 6.6 and 6.7).

While the lack of modelled noise is unrealistic when compared to real data, this simulated data serves to illustrate the limitations of the model (limitations which are applicable to the analysis of shared peptides as a whole) on ideal data with minimal noise.

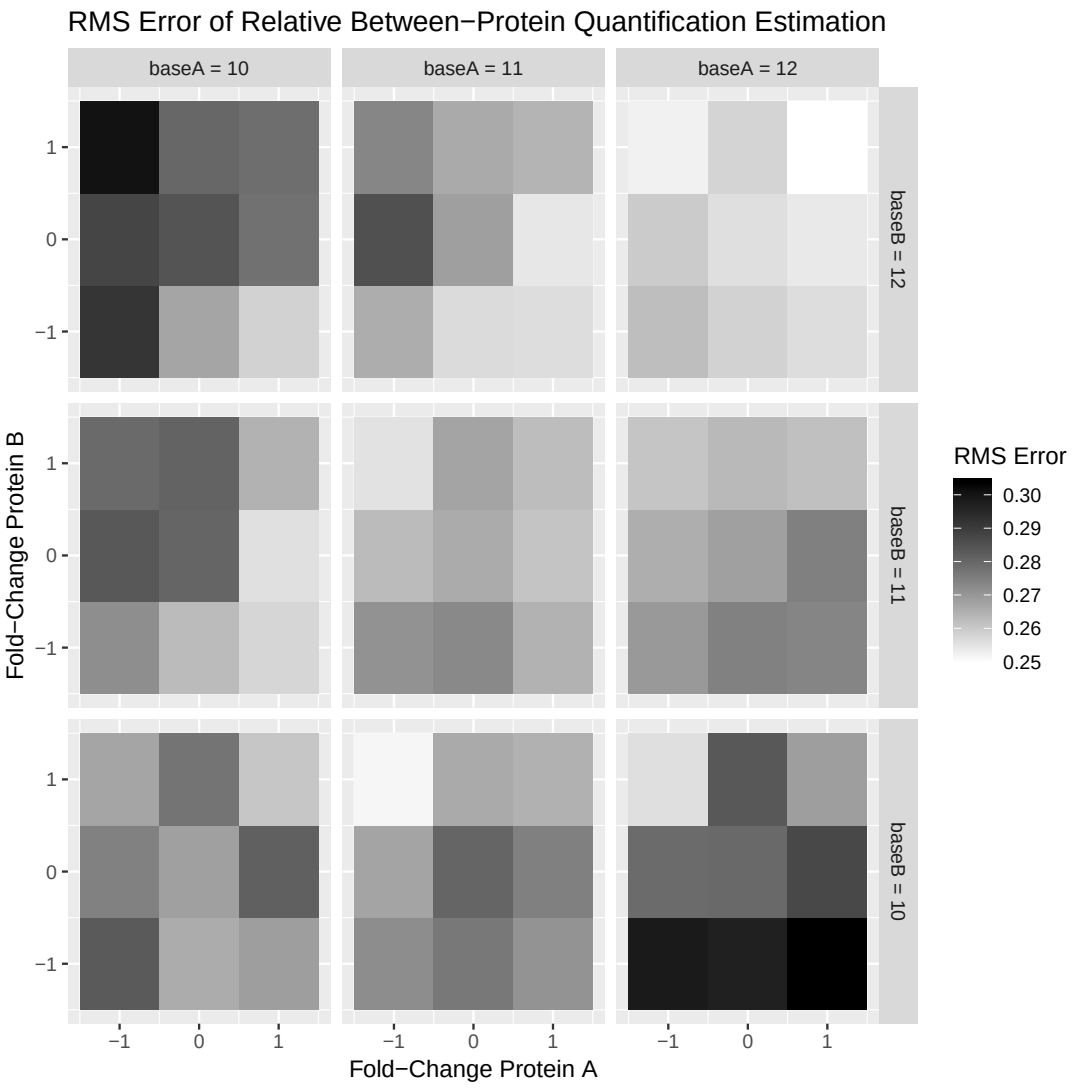

Figure 6.9: Heatmaps showing the RMS error in estimation of relative quantification between proteoforms A and B for the case described in Section 6.5.1 with one shared peptide. Larger error is shown in darker tones. Sub-plots are grouped by the baseline intensity of proteoforms A and B on the x and y axes respectively. The x and y axes of each of the subplots correspond to the log-2 fold-change between the two assay groups of proteoforms A and B respectively. The heatmap is roughly symmetrical across the diagonal.

6.5.2 Two Proteoforms with Bridge Proteoform

The relative between-proteoform quantification of a pair of otherwise unrelated proteoforms can be achieved with the addition of a third bridge proteoform which shares peptides with the two original proteoforms, as illustrated in Figure 6.10

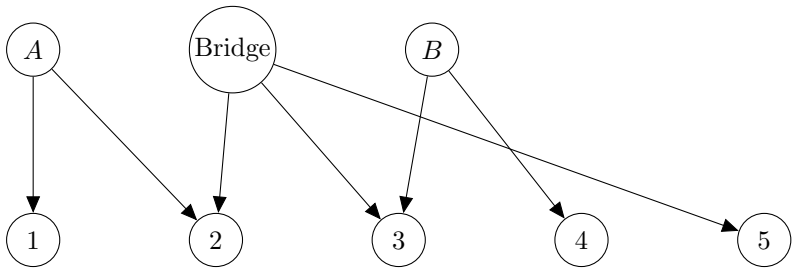

Figure 6.10: Example of a bridge proteoform being added to allow for relative quantification of otherwise unrelated proteoforms. The bridge proteoform is constructed to share peptides 2 and 3 with proteoforms A and B respectively.

Multiple simulated data sets similar to the above were generated. Each peptide had three feature simulated, with their ionisation coefficients sampled from Dirichlet distributions. Sample variance and feature variance were set to zero. Finally, Poisson noise was applied to generate the simulated counts.

The simulated data sets were varied in terms of: the $\log_2$-abundance of the bridge proteoform, $\text{Bridge}_{\text{Ctrl}}$; and the $\log_2$-fold-change of the bridge proteoform between the two sample groups, $\text{Bridge}_{\log_2\text{-Fold-Change}}$. The data sets were simulated using values:

$$\text{Bridge}_{\text{Ctrl}} = 9, 10, 11, 12, 13 \tag{6.54}$$

$$\text{Bridge}_{\log_2\text{-Fold-Change}} = 0, 0.25, 0.5, 1.0, 2.0 \tag{6.55}$$

The shared peptide model was run for each of the 25 simulated data sets. As above, the RMS error in estimating the relative between-proteoform quantification between proteoforms A and B was calculated. The crucial difference between this and the simulated data above is that the relative quantification between proteoforms A and B can only be inferred indirectly through the bridge proteoform. A heatmap of the RMS error is presented in Figure 6.11.

Similar to the simulated data with just two proteoforms sharing a single peptide, in this scenario, a bridge proteoform with a significantly higher abundance than the two proteoforms in question (the top-right of Figure 6.11) results in higher error in

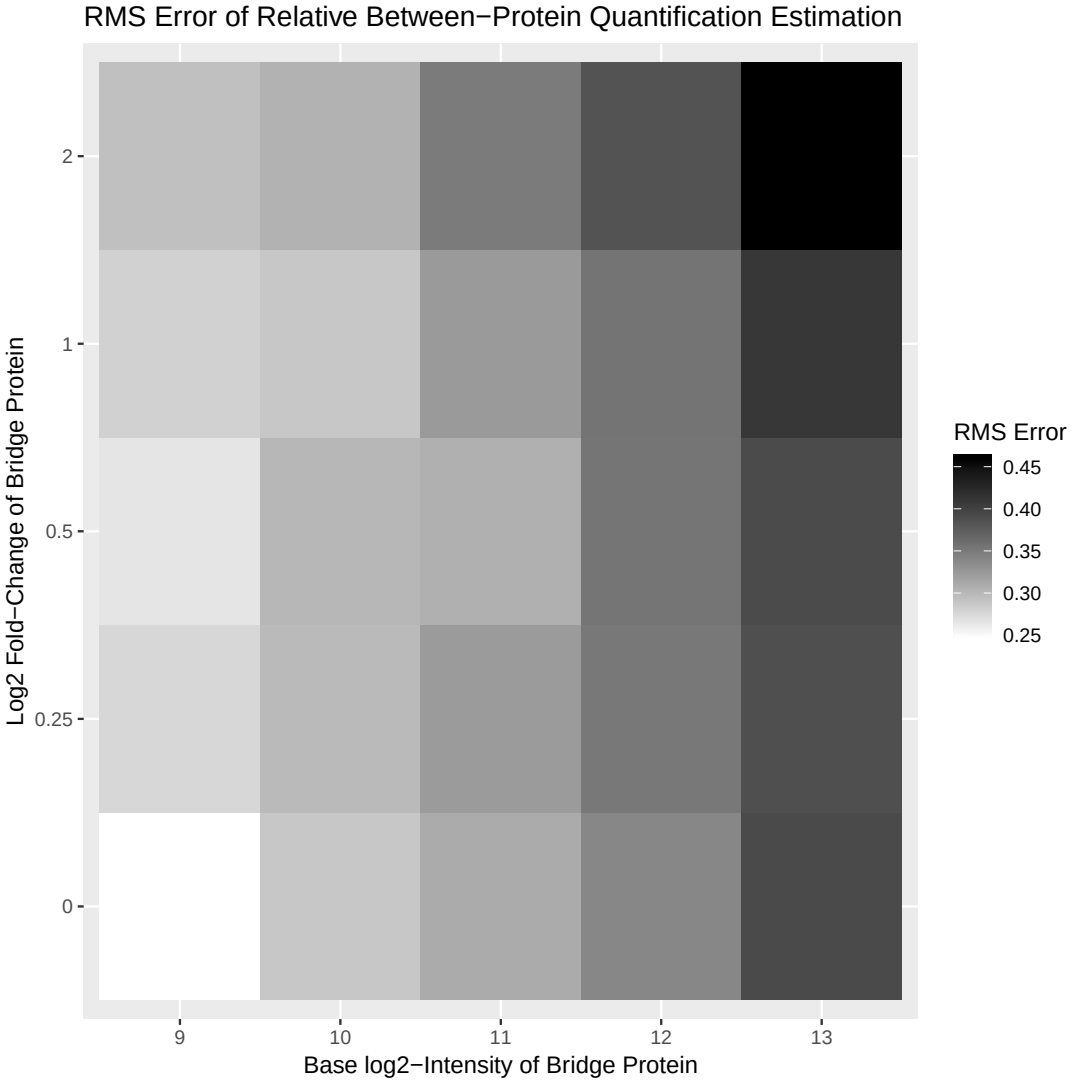

Figure 6.11: Heatmaps showing the RMS error in estimation of relative quantification between proteoforms A and B via a bridge proteoform. Larger error is shown in darker tones. The $\log_2$ abundances of proteoforms A and B were simulated as 10.0 and 12.0 respectively.

the estimation of the relative quantification between the two proteoforms. Again, this is due to the relatively small signal contributed to the two shared peptides by the proteoforms of interest compared with that contributed by the high-intensity bridge proteoform.

The mean error in estimation of the relative quantification between proteoforms can also be visualised. Figure 6.12 shows a heatmap of the mean error in estimating the relative quantification between proteoforms A and B across all eight simulated assays.

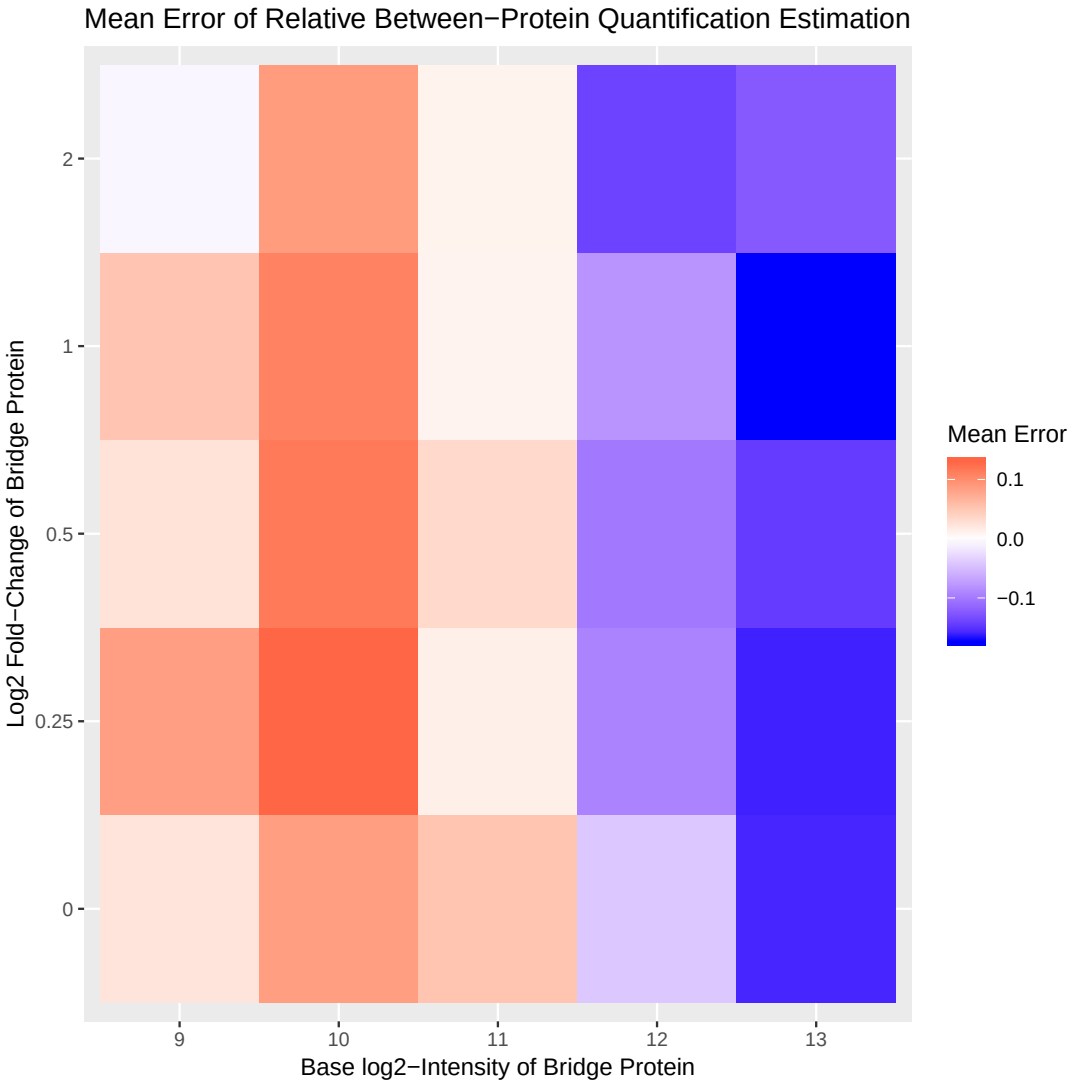

Figure 6.12: Heatmap of the mean error in estimation of relative quantification between two proteoforms with through a bridge proteoform. Positive bias is shown in red tones and negative bias in blue tones.

As in Section 6.5.1, the model should not be able to infer the relative abundances of A and B in the case where the $\log_2$-fold-change of the bridge proteoform is 0.0. Also the model shows some bias in estimating the relative abundances of the proteoforms. Inspection of the posterior samples suggests some non-identifiability for these simulated data sets.

As an illustrative example, Figure 6.13 shows violin plots of the inferred $\log_2$-abundances and relative $\log_2$-ratios of the three simulated proteoforms for the case where the bridge proteoform $\log_2$-intensity is 11.0 and the simulated $\log_2$-fold-change of the bridge proteoform is 0.0.

As in the case with two proteoforms with one shared peptide, the shared peptide between proteoform A and the bridge proteoform allows for the relative abundance of proteoform A and the bridge proteoform to be inferred. Likewise, the shared peptide between proteoform B and the bridge proteoform allows the relative abundances of proteoform B and the bridge proteoform to be inferred. As a consequence, the relative abundance between A and B is inferred as:

$$\text{Rel}_{B:A} = \frac{\text{Rel}_{B:Bridge}}{\text{Rel}_{Bridge:A}} \tag{6.56}$$

Violin plots of the inferred $\log_2$-ratios between the three simulated proteoforms are presented in Figure 6.14.

It can be observed in Figure 6.14 that the $\log_2$-ratio between proteoforms A and B can be inferred indirectly. The true $\log_2$-ratio is 2. Averaging all of the MCMC samples for the estimated $\log_2$-ratio between the two proteoforms across all eight simulated assays, the mean estimated $\log_2$-ratio between proteoforms A and B is 2.05, close to the true value; the RMS error of the samples is 0.311.

Proper validation of this proposed method for relative quantification of unrelated proteoforms would require a specially crafted data set with known concentrations of proteoforms created synthetically, for example, with QConCATs[124]. A proof of concept example found in the spike-in data with similar properties is presented in the Section 6.6.

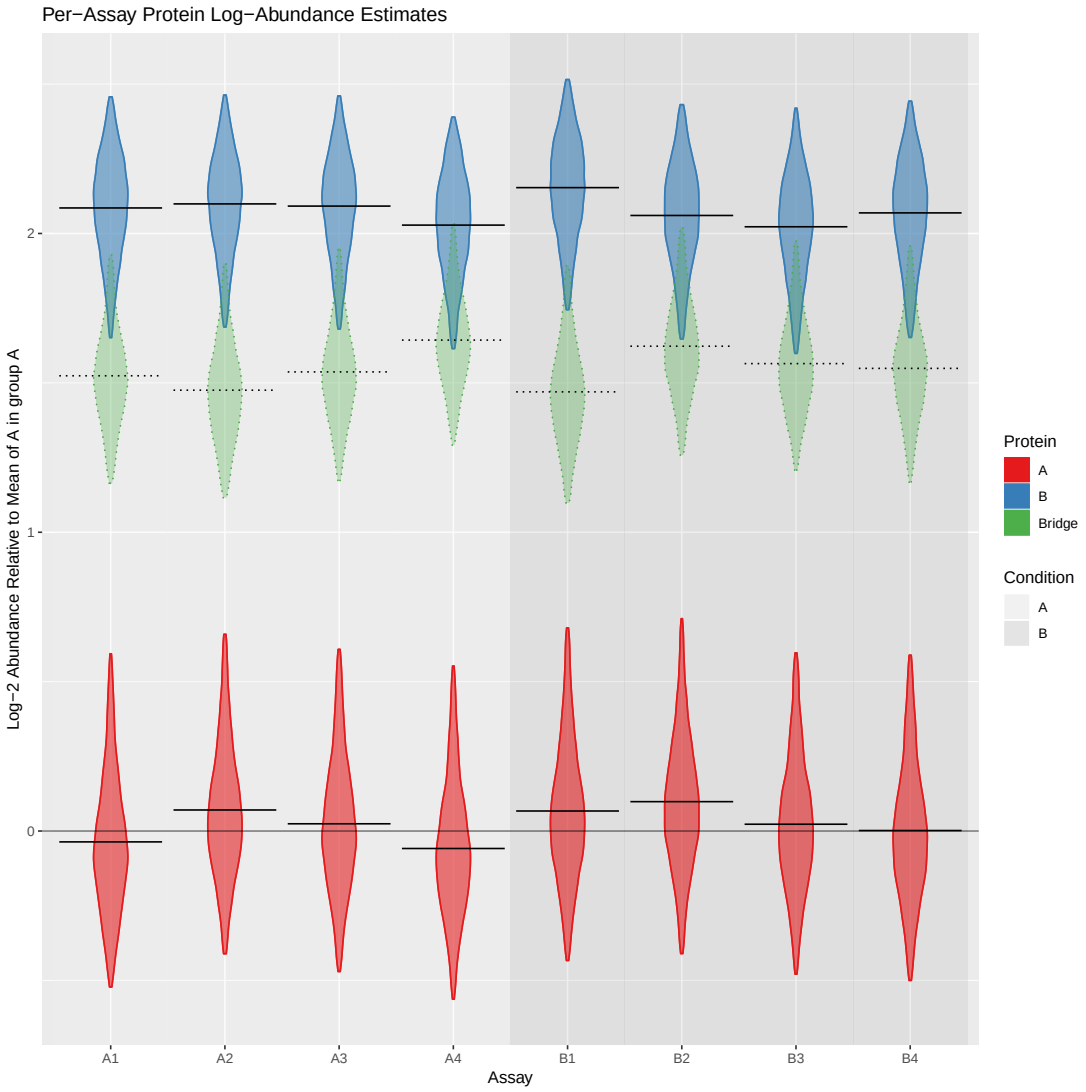

Figure 6.13: Violin plots of inferred $\log_2$-abundances of the three simulated proteoforms, A, B and Bridge. The true $\log_2$-abundances of A and B were simulated as 10.0 and 12.0 respectively. The true $\log_2$-abundance of the bridge proteoform was simulated as 11.0 The y-axis is rescaled relative to the mean $\log_2$-abundance of proteoform A in group A.

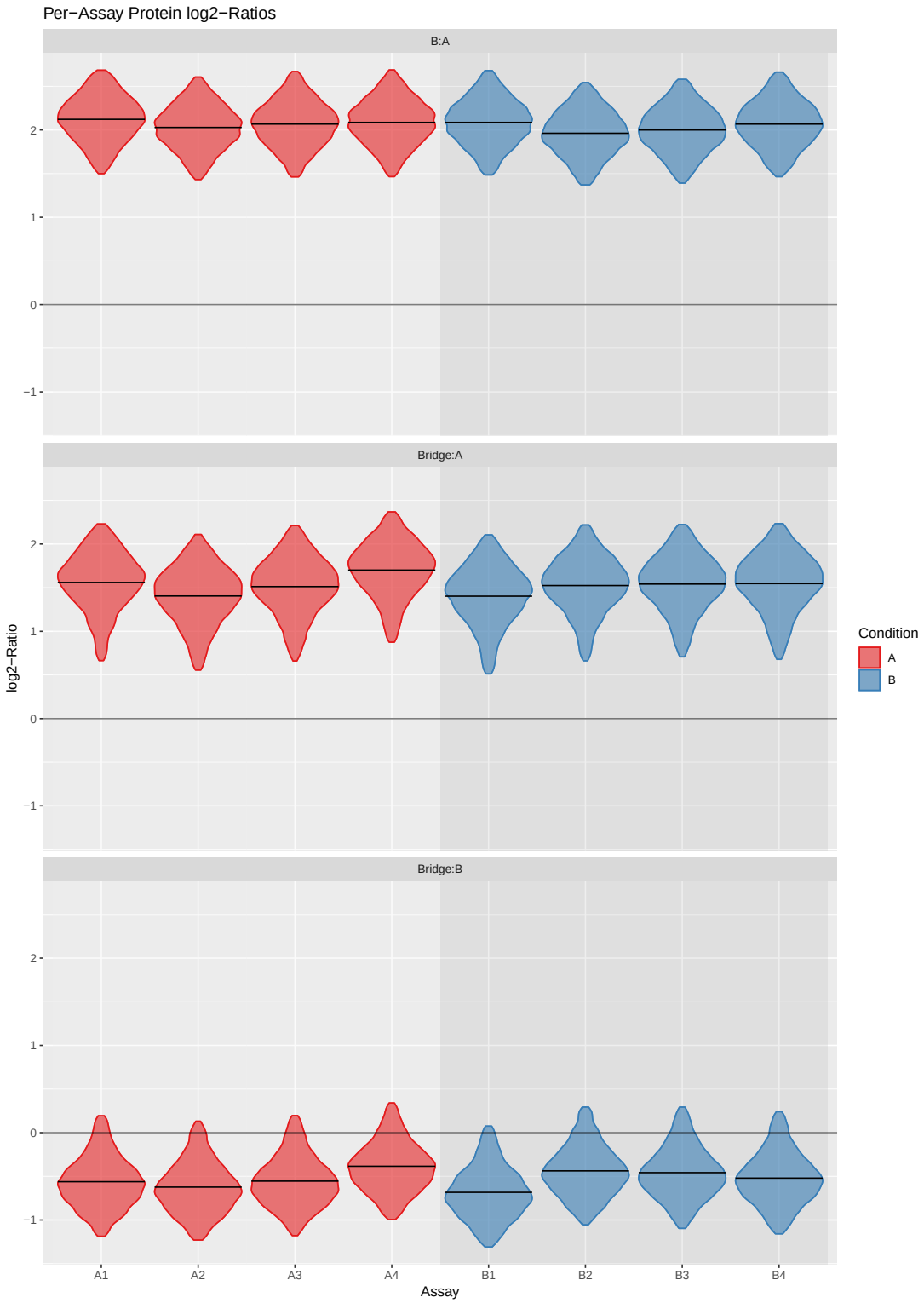

Figure 6.14: Violin plots of the inferred relative $\log_2$-ratios between the three proteoforms, A, B and Bridge in each of the eight simulated assays. The true $\log_2$-abundances of A and B were simulated as 10.0 and 12.0 respectively. The true $\log_2$-abundance of the bridge proteoform was simulated as 11.0

6.5.3 Non-identifiability Problems

It was observed above that the shared peptide model could give biased estimates for the between-proteoform relative quantifications for simulated datasets. In order to illustrate the reasons behind this, a simplified version of the shared peptide model that exhibits the same multiplicative non-identifiability was used.

Two proteoforms, A and B, were simulated with true $\log_2$-abundances of 10 and 12 respectively in two conditions with a $\log_2$-fold-change of zero between them, i.e. both proteoforms were unchanging between the two groups. Each proteoform had one unique peptide and a peptide shared between them and their $\log_2$-abundances $Q_{1:3}$ calculated as the Log-Sum-Exp of the parent proteoform $\log_2$-abundances. Each peptide had a single observable feature simulated, with ionsiation coefficients $I_{1:3} = (1.0, 0.25, 1.0)$ respectively: the ionisation coefficients were chosen so that the resulting intensities of the unique peptides' features were equal, so any inference about the relative abundance of the two proteoforms can only come from the shared peptide. The $\log_2$-intensities y of each feature in the two conditions were then calculated:

$$y_i = \log_2(I_i) + Q_i \tag{6.57}$$

The observed $\log_2$-intensities of the features were then used as input data for the simplified model and the model was fit using Stan to infer the $\log_2$-abundances of proteoforms A and B in the two groups.

Figure 6.16 shows a plot of the MCMC samples for $\log_2$-abundance of proteoform B versus the $\log_2$-abundance of proteoform A in the control group. Where there is ambiguity in the "correct answer", the multiplicative non-identifiability results in a posterior with some banana-shaped geometry. Marginalising to obtain the posterior for the relative between-proteoform quantification, the resulting marginal posterior is bimodal in this scenario, with modes at the two most likely solutions. Figure 6.17 shows a density plot of this inferred relative abundance.

It has been established[125] that MCMC samplers can struggle to sample effectively from posteriors with awkward geometry such as the banana-like posterior that has the potential to appear in the shared peptide model due to the non-identifiability. Similar issues arise in models with hierarchical variance parameters, such as the well-documented cases of Neal's funnel[126] and the Eight-schools problem[127], where failure to explore the posterior properly can result in biased inference. Here the solution is often to reparameterise the model so that the posterior can be sampled from more effectively[123]: a reparameterisation for the model for shared peptide analysis is not

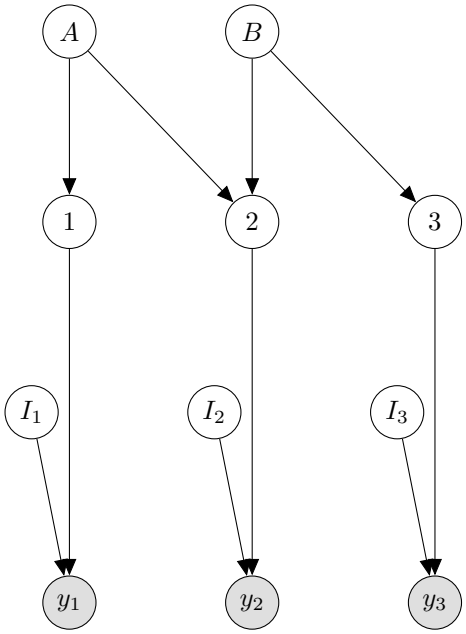

Figure 6.15: Bayesian Network Diagram of the Simplified model to Demonstrate Non-identifiability. Two proteoforms A and B produce unique peptides 1 and 3 and a single peptide 2 shared between them. Only variables for a single treatment group are shown.

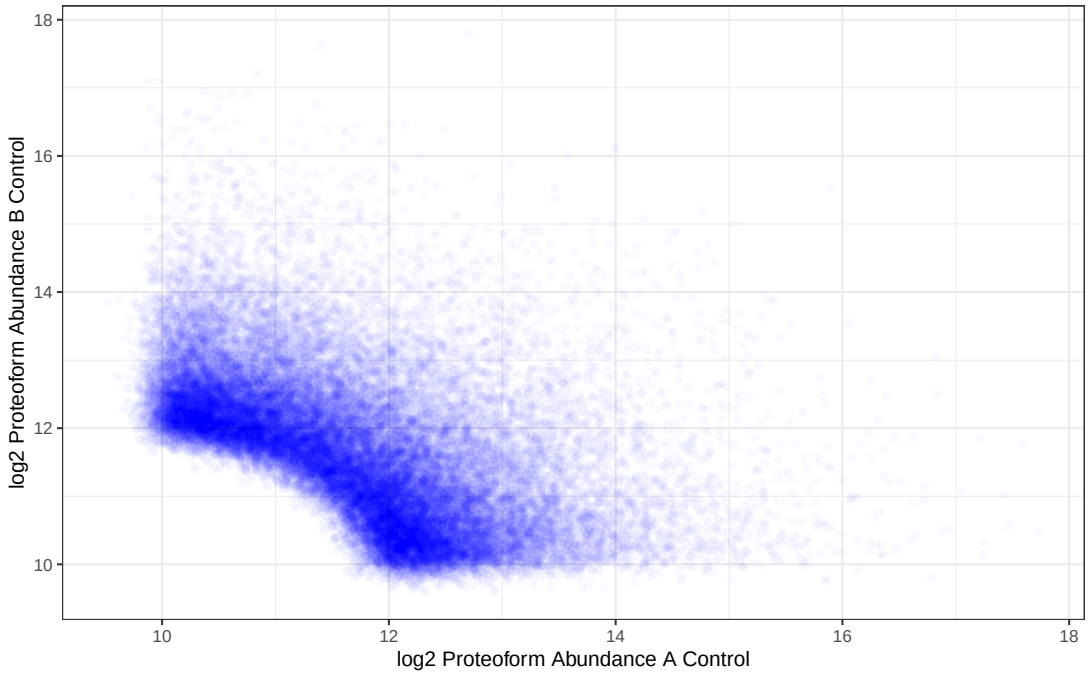

Figure 6.16: Plot of the Inferred $\log_2$-abundance of Proteoform B versus Proteoform A in the Control Group for the Simplified Shared Peptide Model. Each point represents an MCMC sample. The multiplicative non-identifiability from the ambiguous data results in a banana-like posterior density.

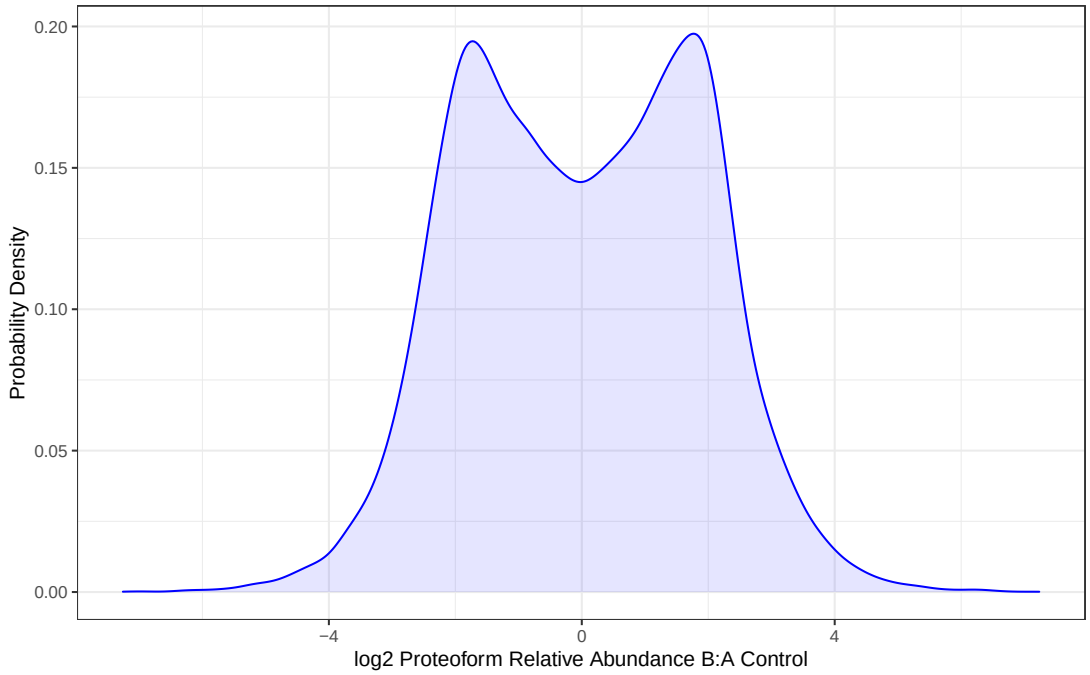

Figure 6.17: Density Plot of the Inferred Relative $\log_2$-abundance between Proteoform B and Proteoform A for the Simplified Shared Peptide Model. Marginalising the banana-shaped posterior from Figure 6.16 results in a bimodal posterior for the relative $\log_2$-abundance.

immediately apparent. In the absence of a suitable reparameterisation the application of a numerical integrator with adaptive stepsize or sequential Monte Carlo methods are two possible solutions to this problem.

6.6 Results from Spike-In Data

Results from a sample of the spike-in data set used to validate the BayesProt algorithm in Chapter 3 are presented.

Since many of the proteins which were individually spiked-in are mammalian (as is the rat background) and species variants of the functionally equivalent proteins are relatively similar in sequence, this leads to significant numbers of shared peptides.

One of the simpler examples in the data set is a related pair of cytochrome proteins, one rat, "sp|P62898|CYC_RAT" and one horse, "sp|P00004|CYC_HORSE". The rat protein forms a constant baseline and the horse protein was spiked at two levels.

Most of their amino acid sequence overlaps and, as a consequence, there are four peptides that are shared between the two proteins. The horse protein has six unique peptides and the rat protein has four unique peptides.

Analysing CYC_HORSE and CYC_RAT Separately

As a baseline, each protein is analysed separately using only the unique peptides for each, excluding the shared peptides. Violin plots of the inferred assay-level effects are presented in Figures 6.18 and 6.19.

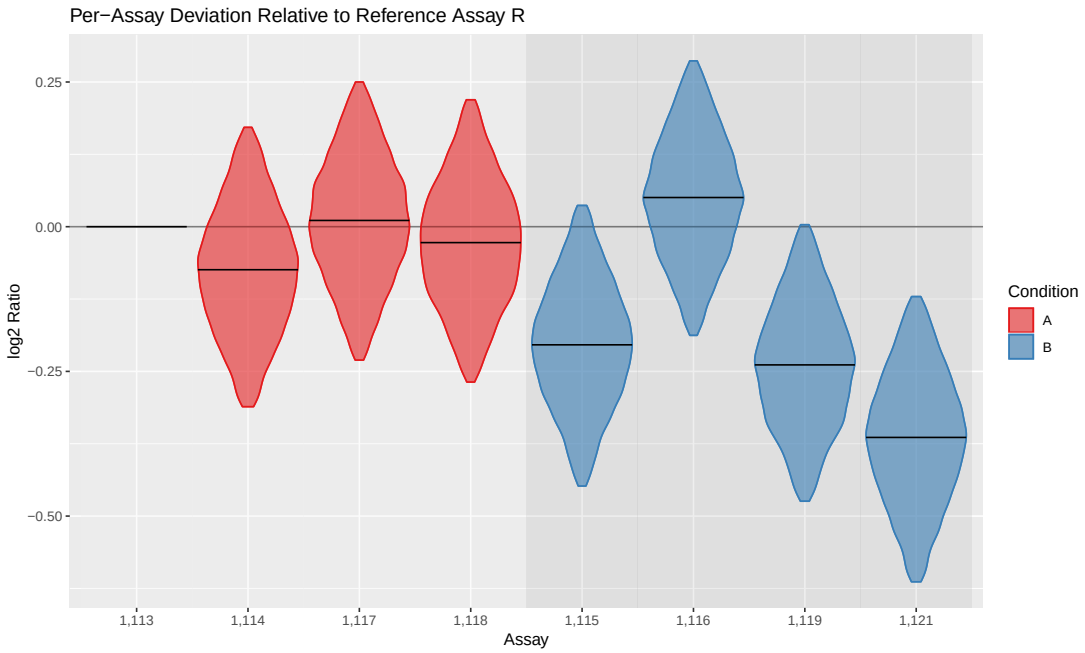

Figure 6.18: Sample effects of CYC_RAT using only four unique peptides

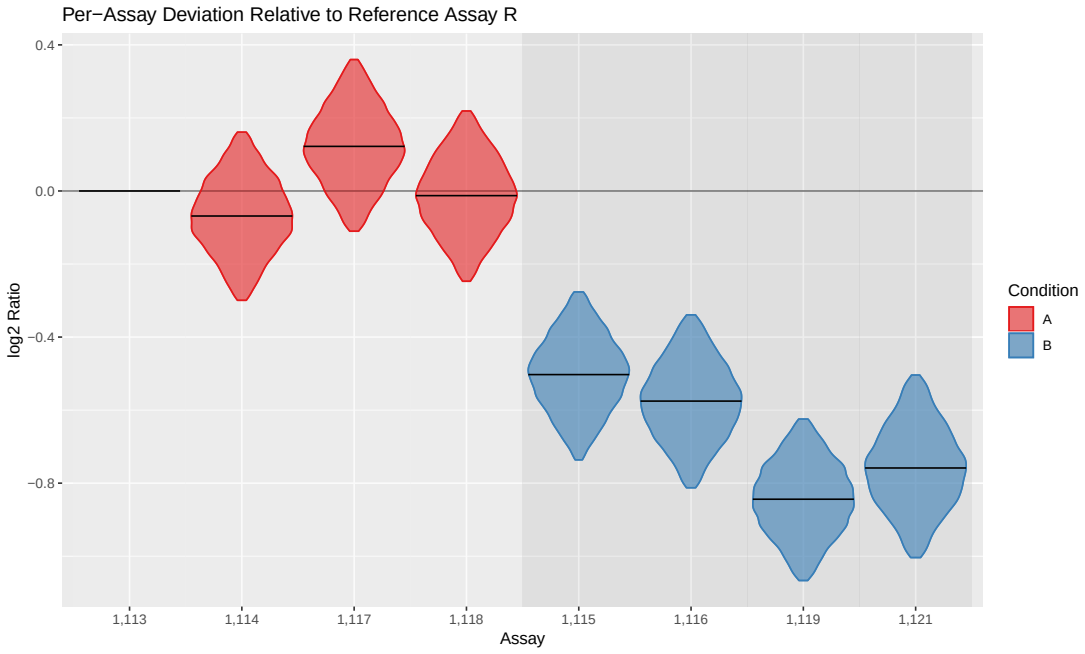

Figure 6.19: Sample effects of CYC_HORSE using only six unique peptides

Analysing CYC_HORSE and CYC_RAT with Shared Peptides

The shared peptide model was then applied to the combined data for the CYC_HORSE and CYC_RAT proteins, including the four shared peptides. Violin plots of the inferred assay-level effects and $\log_2$-abundances are presented in Figures 6.20 and 6.21.

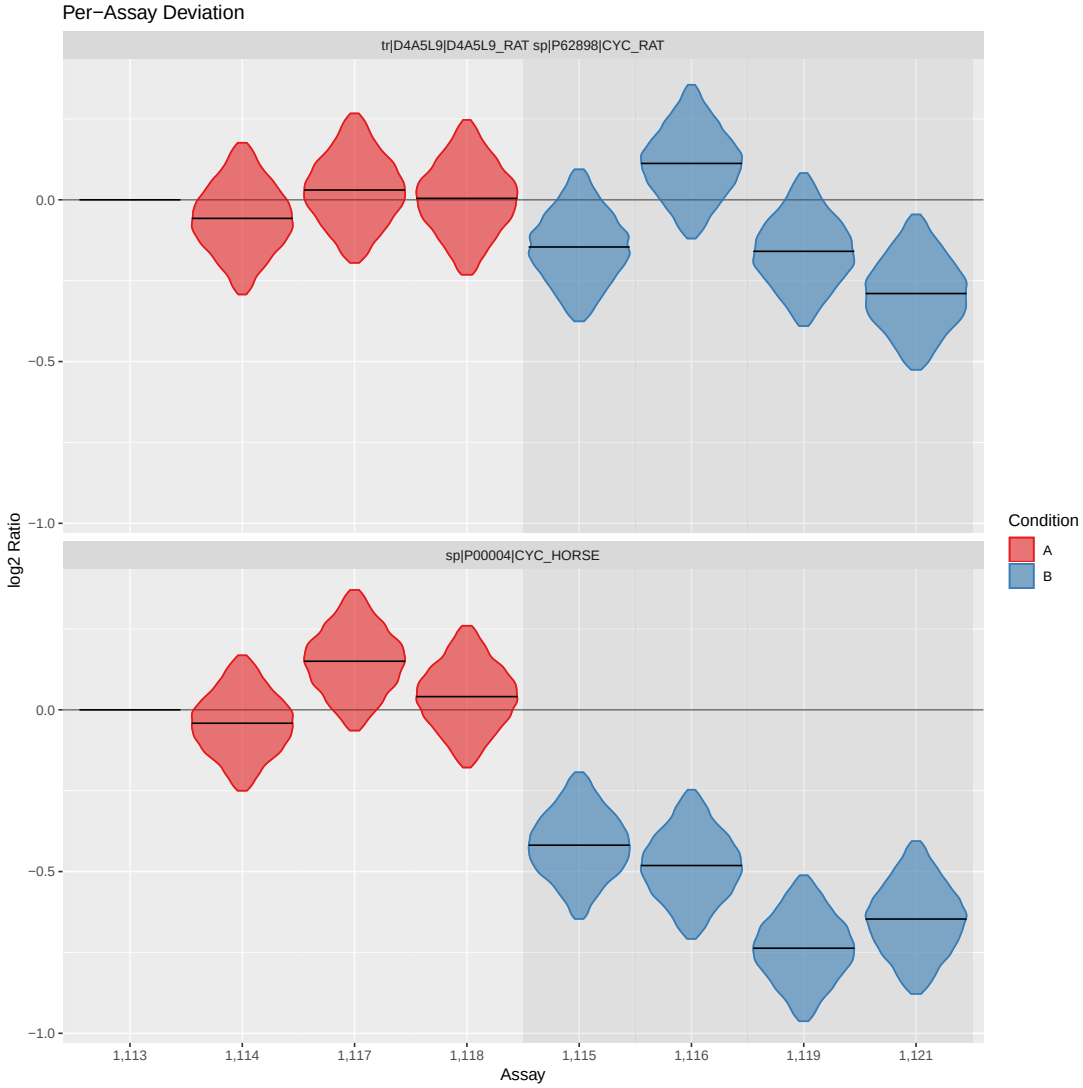

Figure 6.20: Per-assay effects of CYC_RAT and CYC_HORSE using all available peptides, including the four shared peptides.

The Bayesian model comparison method developed in Chapter 5 is used to perform statistical testing on the inferred sample quantifications from the models both with and without shared peptides so that the effect on differential expression testing can be

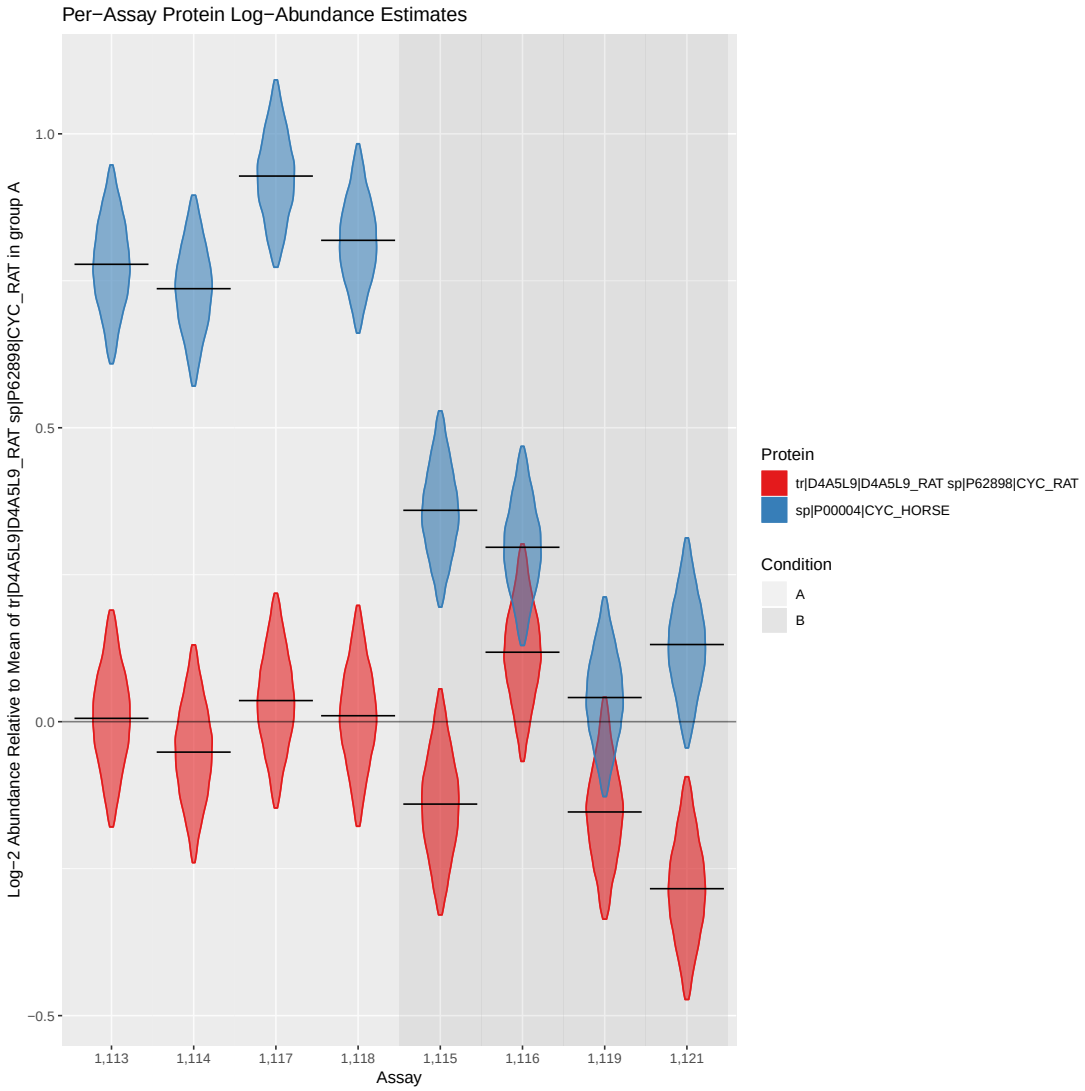

Figure 6.21: Inferred protein $\log_2$-abundances of CYC_RAT and CYC_HORSE

Table 6.1: Statistical testing of the CYC_RAT and CYC_HORSE proteins with and without shared peptide modelling.

	PEP	
Protein	Naïve Model	Shared Peptide Model
CYC_RAT	0.954	0.975
CYC_HORSE	0.459	0.000 123

examined. The median and median absolute deviation of the MCMC samples for each assay are used to summarise the MCMC samples, which are then used as a measure of the location and uncertainty of the quantification estimates, as in Chapter 5. The prior parameters were set using the empirical Bayes procedure used previously on the whole spike-in data set in Chapter 5. The results of these tests are presented in Table 6.1.

The inclusion of shared peptides in the modelling process allows for the differential expression of the spiked-in horse protein to be better discerned; the posterior error probability of differential expression falls into a range where it is much more likely to be included in a candidate list of proteins. Additionally, the posterior error probability of the rat protein increases, suggesting increased confidence that the rat protein is not differentially expressed. The model has the added ability to quantify the relative abundances of proteins (or proteoforms) to some degree.

6.6.1 Bridge Protein Example

A convenient example of a bridge protein was found within the spike-in data set, with two rat proteins with SwissProt accessions "sp|P23965|ECI1_RAT" and "sp|Q63481|RAB7L_RAT" sharing a single peptide, "VLVEK". This peptide was also shared with an E. coli protein with SwissProt accession "sp|P0AED0|USPA_ECOLI". Each of the three proteins had unique peptides associated with them.

From these estimates of the log-abundance of each protein in each sample, we can calculate the relative difference in abundances for each pair of proteins, which are depicted in Figure 6.23.

The Bayesian model is able to infer the relative ionisation weights of each feature in the data from the ionisation of the features of the E. Coli protein. This scenario differs slightly from the simulated one analysed above in that the shared peptide linking the three proteins together is shared by the rat proteins. However, the simulated example motivates the argument that a carefully constructed set of QConCAT[124] proteins could provide absolute quantification across an entire proteome if spiked-into two sets

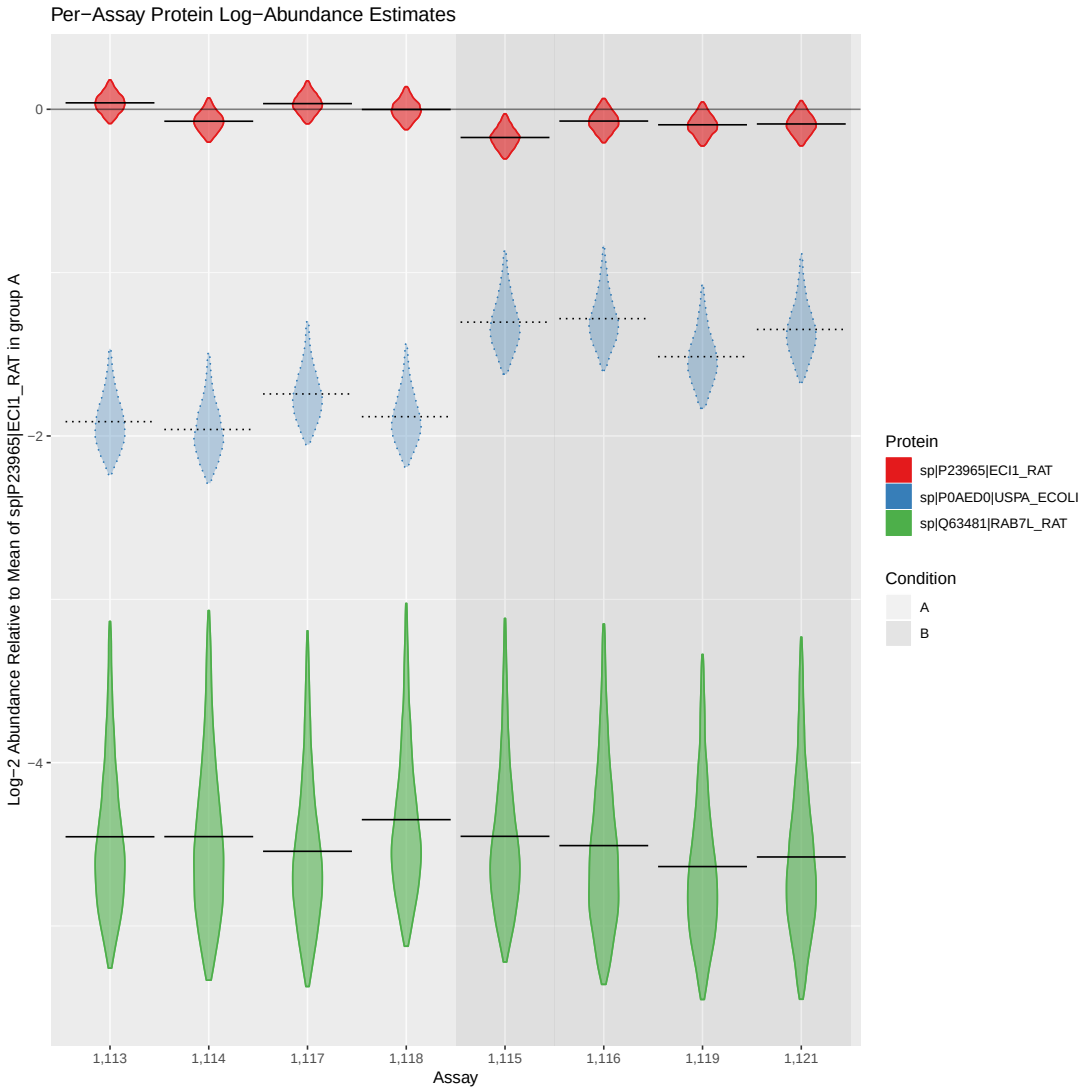

Figure 6.22: Inferred protein $\log_2$-abundances from bridge protein example.

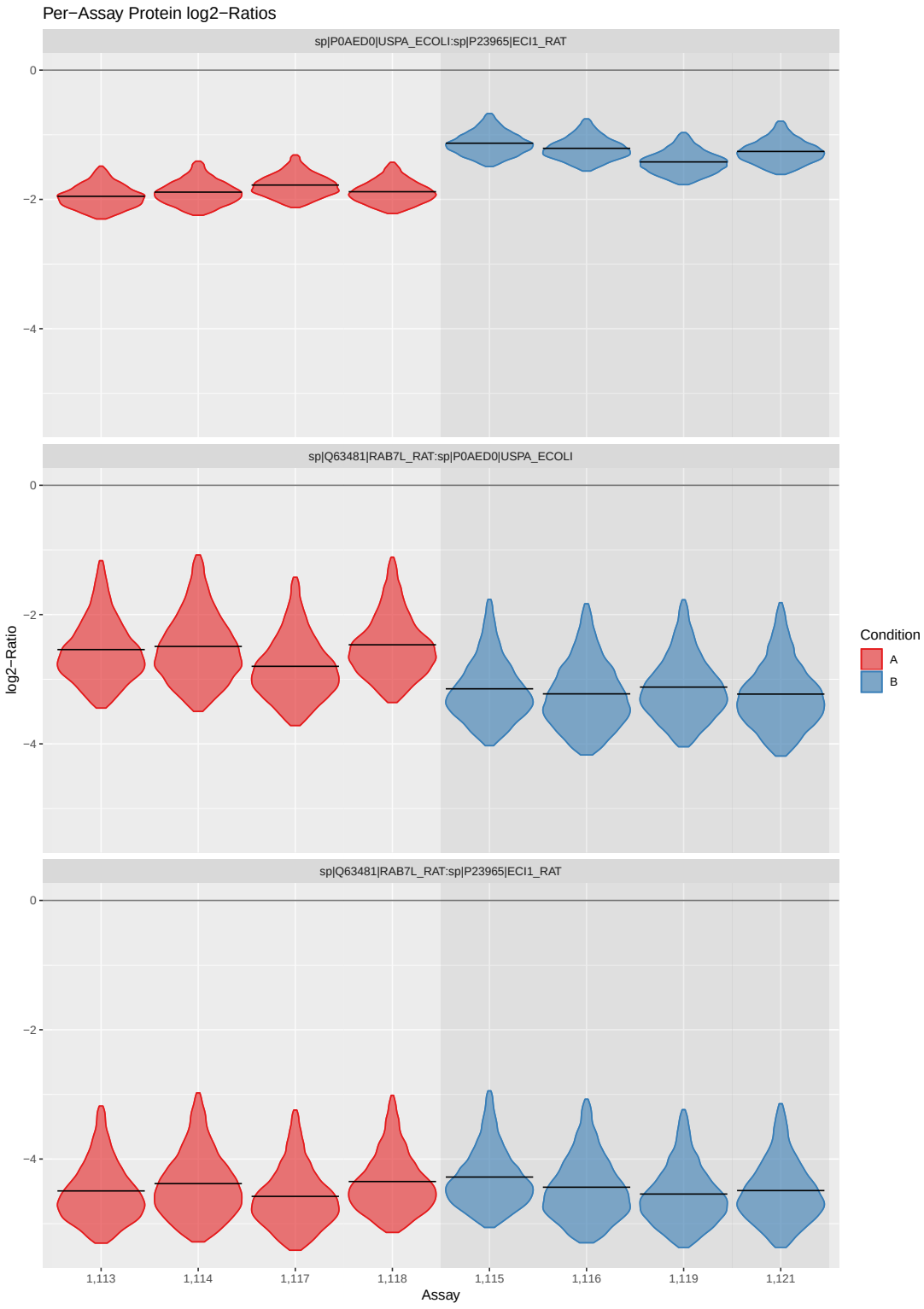

Figure 6.23: Per-assay inferred relative between-protein quantifications from bridge protein example.

of samples.

6.7 Results on Amyloid Beta Peptide Data

Further to the above results on both simulated and spike-in data, the effectiveness of a shared peptide model on a small subset of a clinical study is demonstrated. An iTRAQ quantification study[2] was performed comparing post-mortem biopsy samples from several brain regions in Alzheimer's disease (AD) affected cases, comparing against age- and gender-matched controls.

The amyloid beta peptides have long been associated with the development of Alzheimer's disease, and the presence of amyloid plaques is a key marker in post-mortem study of AD-affected brains. Hence, understanding how the expression of these peptides in different brain regions relates to disease severity is important to Alzheimer's research.

The amyloid beta peptides's presence in mass spectrometry data is difficult to discern since it is proteolysis product of much larger amyloid precursor protein (APP), which is around 770 amino acids in length.

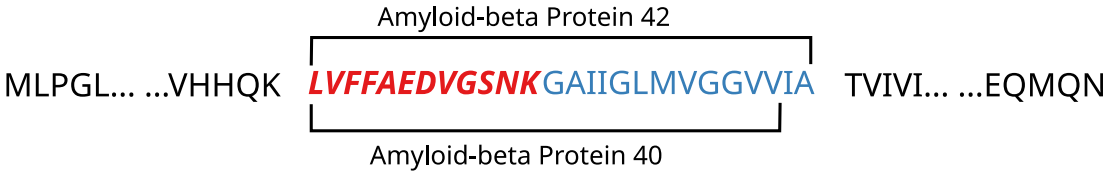

Figure 6.24: Amyloid beta protein sequences shown as substrings of amyloid beta precursor protein. The tryptic peptide visible in the data from [2] is highlighted in red. Since it is a substring of both Abeta-40 and Abeta-42, the two amyloid beta proteins are indistinguishable in the data.

Abeta-40 (sequence: "DAEFRHDSGYEVHHQKLVFFAEDVGSNKGAIIGLMVGGVV") and Abeta-42 (sequence: 'DAEFRHDSGYEVHHQKLVFFAEDVGSNKGAIIGLMVGGVVIA") are indistinguishable in the data[2], since the peptide produced by digestion is "LVFFAEDVGSNK", a substring of both (see Figure 6.24). Hereafter, amyloid beta or Abeta will refer to the sum of signals of these two peptides. The signals associated with an Abeta peptide existing in isolation are indistinguishable from signals from the digested sequence of APP.

6.7.1 Experimental Design

Six brain regions were analysed:

- Hippocampus (HP)

- Motor cortex (MCx)
- Sensory cortex (SCx)
- Cerebellum (CB)
- Entorhinal cortex (ENT)
- Cingulate gyrus (CG)

Tissue samples from the six regions were collected from the brains of eighteen subjects: nine AD-affected patients (S1–S9) and nine age- and gender-matched controls (S10–S18). For each brain region a pooled reference sample, R, was created by mixing equal amounts of each of the eighteen samples together. Each region was processed as a separate experiment of three iTRAQ 8-plexes. The experimental design is described in Table 6.2.

The data from each brain region was fitted with two models as above; treating APP and amyloid beta as separate proteins and with amyloid beta as a subset protein of APP using the shared peptide model.

For each brain region, for both the shared peptide model and the model treating the two proteins separately, eight MCMC chains were run each with 200,000 samples of which the first 100,000 were discarded as burn-in. Samples were then thinned, keeping every 20th sample, resulting in 5000 samples being kept from each chain. This was for the convenience of saving disk space and for memory usage creating the subsequent plots.

For conciseness, the results from the cingluate gyrus and sensory cortex regions are presented here to illustrate the effect of including shared peptides in the analysis over a naïve analysis which treats the Abeta peptide as a totally separate protein, demonstrating the shared peptide model's potential to quantify the relative abundance of proteoforms or subset proteins in a real scenario.

Table 6.2: Experimental design of the iTRAQ experiments for the Alzheimer's Disease study in [2].

iTRAQ Run	iTRAQ Channel	Assay	Sample	Condition
A	113	R	R1	Pool
A	114	R	R2	Pool
A	115	S1	S1	AD
A	116	S3	S3	AD
A	117	S7	S7	AD
A	118	S12	S12	Ctrl
A	119	S17	S17	Ctrl
A	121	S10	S10	Ctrl
B	113	R	R3	Pool
B	114	R	R4	Pool
B	115	S2	S2	AD
B	116	S6	S6	AD
B	117	S9	S9	AD
B	118	S13	S13	Ctrl
B	119	S15	S15	Ctrl
B	121	S18	S18	Ctrl
C	113	R	R5	Pool
C	114	R	R6	Pool
C	115	S4	S4	AD
C	116	S5	S5	AD
C	117	S8	S8	AD
C	118	S11	S11	Ctrl
C	119	S14	S14	Ctrl
C	121	S16	S16	Ctrl

6.7.2 Naïve Analysis Treating Amyloid Beta as Separate Protein

We first demonstrate a naïve analysis on the amyloid beta and amyloid precursor protein data which treats the amyloid beta peptide as a separate protein. This replicates the analysis published in [2], but with a model in the Stan programming language equivalent to a single protein version of the shared peptide model described above.

Cingulate Gyrus Region

In Figure 6.25, violin plots showing the posterior estimates of the assay effects for the model treating amyloid beta as a separate protein in the cingulate gyrus region are presented. Similarly, Figure 6.26 presents violin plots of the posterior estimates of the assay effects of the APP protein in the cingulate gyrus region.

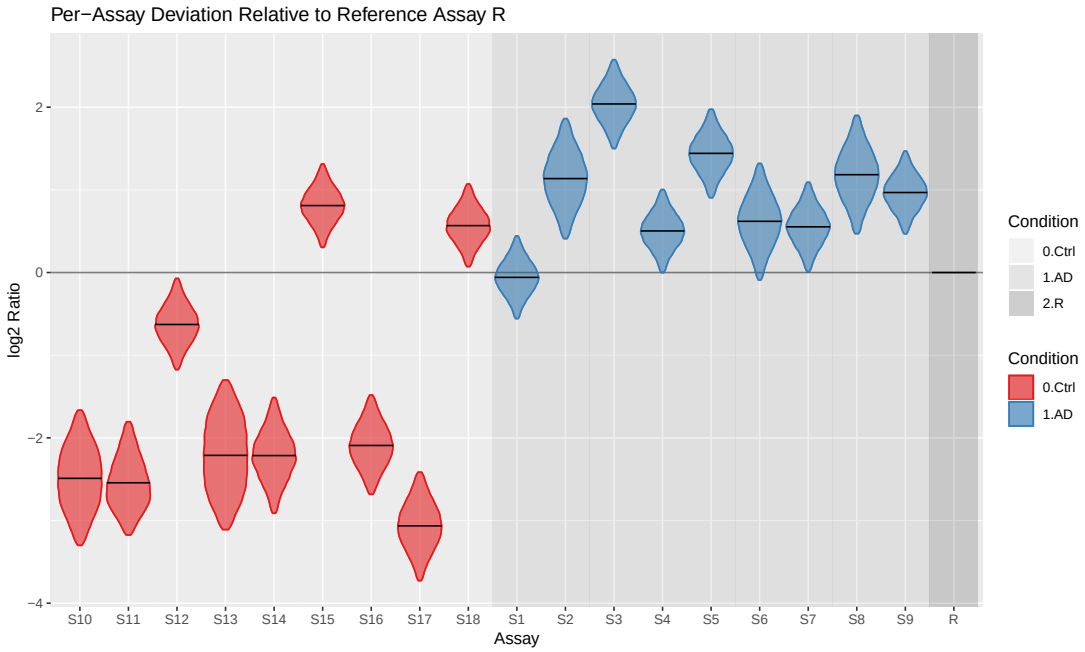

Figure 6.25: Violin plots showing posterior estimates of the assay effect for amyloid beta protein in the cingulate gyrus region when treating amyloid beta as a separate protein.

In the cingulate gyrus region, treating the amyloid beta peptide as a separate protein implies that there is some discernible difference in the abundances between the control and AD samples with the exception of assays S12, S15 and S18. In [2], S15 is noted to have been re-examined and re-classified as a pre-clinical AD case. It is also noted, given the ages of the controls, varying levels of pre-clinical AD may have been present

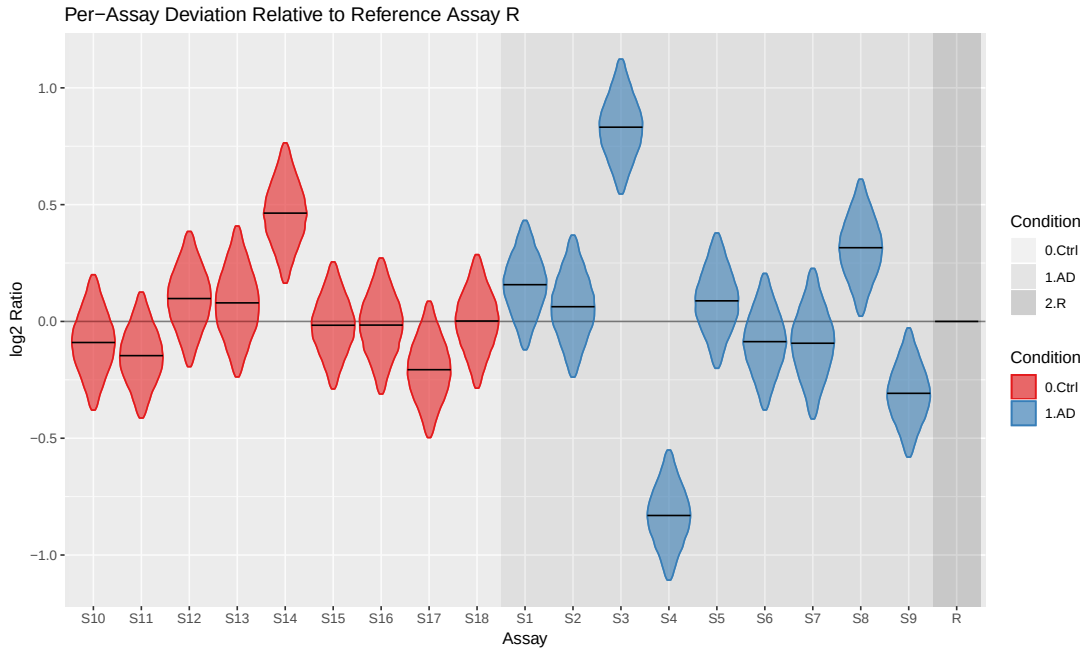

Figure 6.26: Violin plots showing the posterior estimates of the assay effect for the amyloid precursor protein in the cingulate gyrus region when treating it as a separate protein.

in these samples[2]. Hence, it is not unreasonable that S12 and S18 might have been similarly undiagnosed, which would go some way to explain the high levels of the Abeta protein.

Sensory Cortex Region

Figure 6.27 presents violin plots showing the posterior estimates of the assay effects for the model treating amyloid beta as a separate protein in the sensory cortex region. Similarly, in Figure 6.28, violin plots of the posterior estimates of the assay effects of the APP protein in the sensory cortex region are presented.

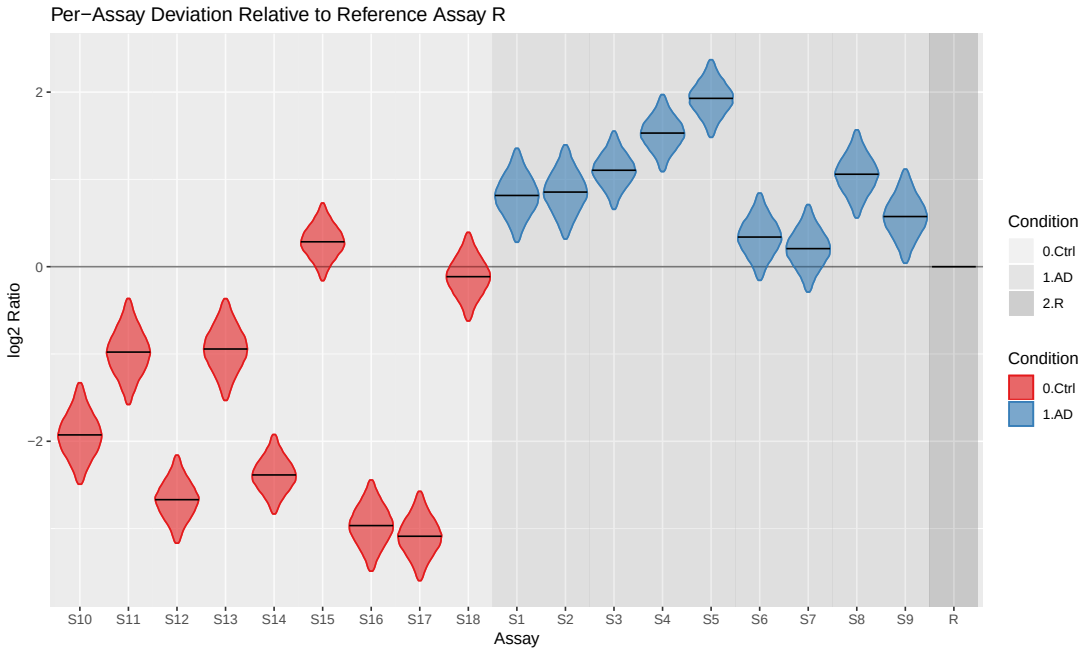

Figure 6.27: Violin plots showing posterior estimates of the assay effect for amyloid beta protein in the sensory cortex region when treating amyloid beta as a separate protein.

As in the cingulate gyrus region, there is a visible increase in the expression of the amyloid beta peptide between the control and AD assays.

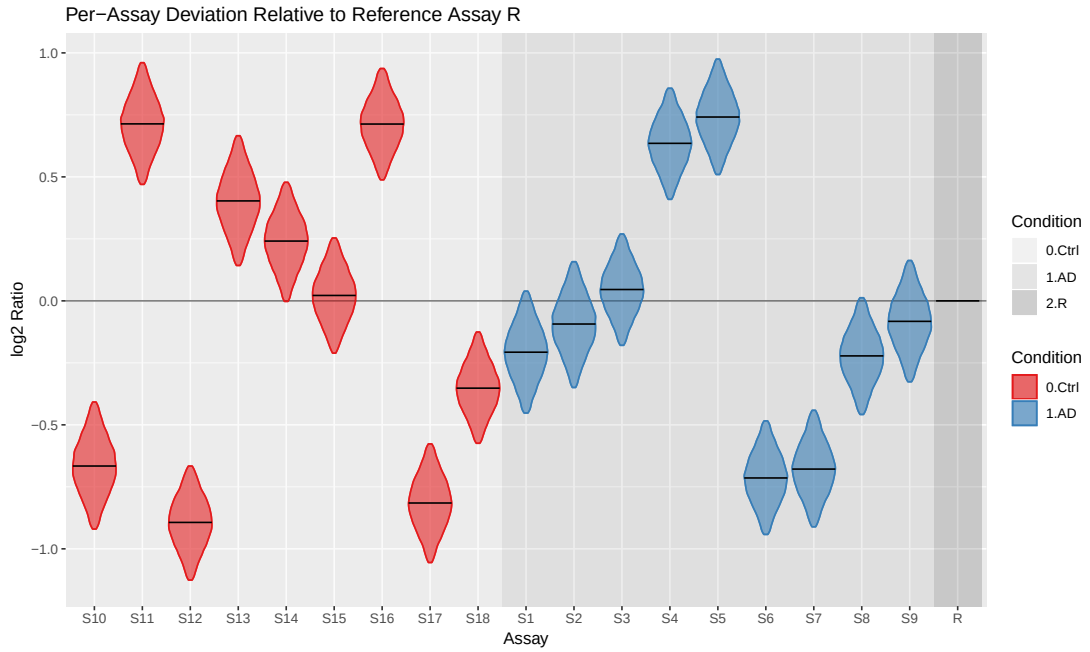

Figure 6.28: Violin plots showing the posterior estimates of the assay effect for the amyloid precursor protein in the sensory cortex region when treating it as a separate protein.

Limitations

There are a few limitations in this simpler analysis. Taking the inferred peptide quantification of the shared peptide as a proxy for the quantification of the amyloid beta peptide fails to take into account the contribution to the quantification by the APP protein. Furthermore, the relative abundances of APP and Abeta cannot be inferred.

6.7.3 Treating Amyloid Beta as a Shared Peptide

For the model including shared peptides, as in previous sections, violin plots of the per-protein assay-level deviations from the reference assay R are presented. Additionally, violin plots of the inferred abundances of the two proteins for each assay are presented.

Cingulate Gyrus Region

Figure 6.29 presents violin plots of the posterior estimates of the assay effects of both APP and Abeta when the shared peptide is correctly modelled. Similarly, violin plots of the posterior estimates of the abundances of APP and Abeta are presented in Figure 6.30.

In Figure 6.30, there is strong evidence that the Abeta protein is differentially expressed between the control and AD samples, excepting some control samples whose levels of Abeta are comparatively high, as noted above. With the shared peptide model, the comparatively low abundance of the Abeta protein in the control samples can be inferred. The ratio of the abundances between the two proteins can be inferred, which is impossible without the inclusion of shared peptides.

Using the Bayesian model comparison method developed in Chapter 5 to perform differential expression testing on the obtained assay effect estimates gives an idea of the effect that the shared peptide modelling would have on the downstream conclusions. Table 6.3 shows the posterior error probabilities produced.

Table 6.3: Statistical testing of the APP and Abeta proteins in the cingulate gyrus region with and without shared peptide modelling.

	PEP	
Protein	Naïve Model	Shared Peptide Model
APP	0.981	0.994
ABeta	0.000 006 23	0.000 228

Here, the shared peptide modelling has a small effect on the posterior error prob-

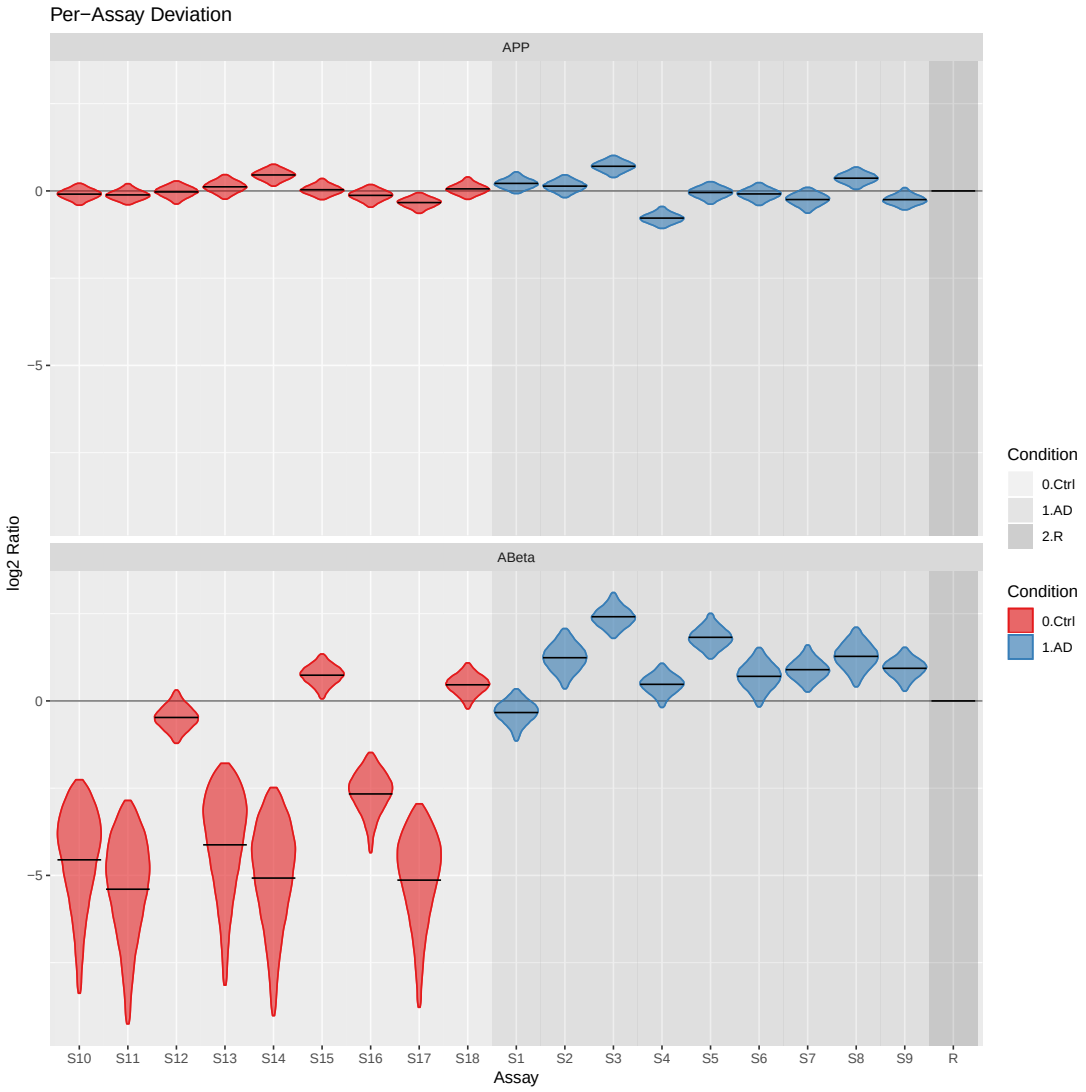

Figure 6.29: Violin plots showing posterior estimates of the assay effect for amyloid precursor protein and amyloid beta protein in the cingulate gyrus region when correctly modelling shared peptides.

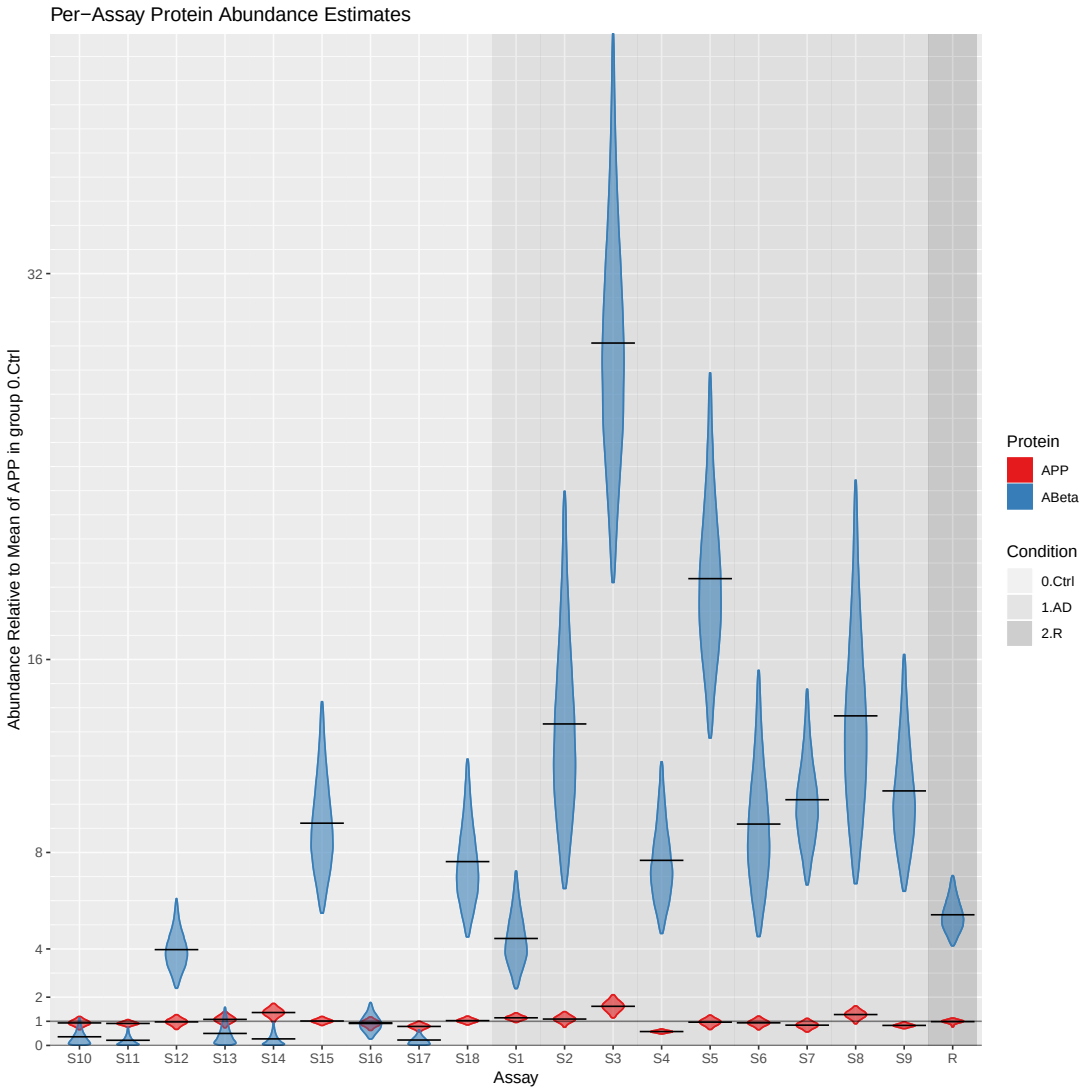

Figure 6.30: Violin plots showing the posterior estimates of the protein-level abundances of the APP and Abeta proteins in the cingulate gyrus region.

ability of the Abeta protein: with and without the shared peptide modelling, there is a high probability of differential expression between the control group and AD group. The posterior error probability of differential expression for the APP protein suggests greater certainty that the APP protein is unchanging between the control and AD group.

Sensory Cortex Region

Figure 6.31 presents violin plots showing the posterior estimates of the assay effects for both APP and Abeta in the sensory cortex region. Violin plots showing the posterior estimates of the abundances of APP and Abeta in the sensory cortex region are presented in Figure 6.32.

Similar to the cingulate gyrus region, in the sensory cortex region, there is evidence of a clear change in Abeta levels between the two groups. As above, the S15 and S18 samples (but not S12) show similar levels of Abeta to the AD group; the rest of the control group have a large amount of their probability mass for the level of Abeta near zero, suggesting its absence. The relative amounts of Abeta and APP can be inferred for each sample. Statistical testing of the sample effects for the sensory cortex region is summarised in Table 6.4.

Table 6.4: Statistical testing of the APP and Abeta proteins in the sensory cortex region with and without shared peptide modelling.

	PEP	
Protein	Naïve Model	Shared Peptide Model
APP	0.981	0.994
ABeta	0.000 000 096 1	0.000 079 1

As in the cingulate gyrus region, there is strong evidence that the Abeta protein is differentially expressed between the two groups, with and without the modelling of shared peptides.

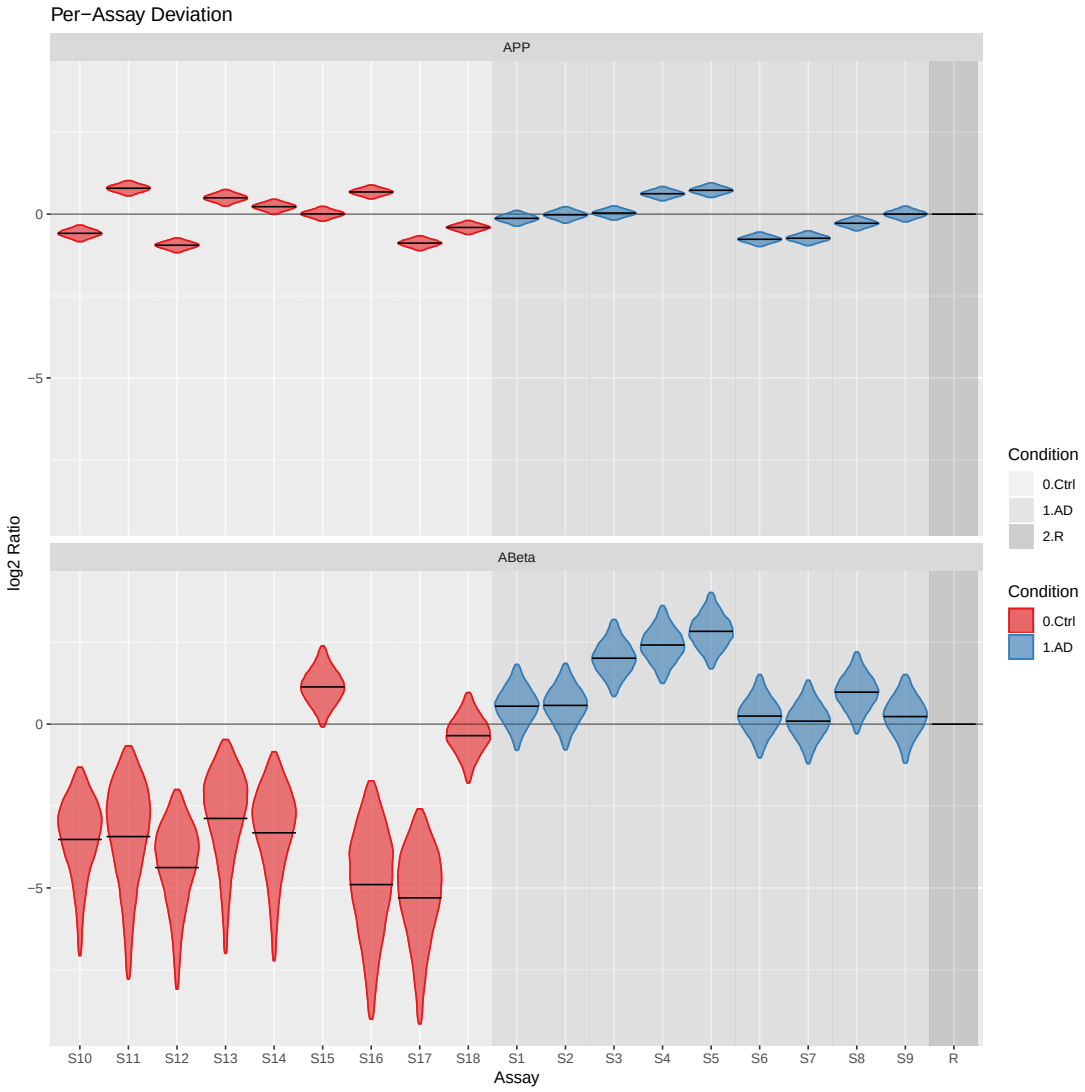

Figure 6.31: Violin plots showing posterior estimates of the assay effects for amyloid precursor protein and amyloid beta protein in the sensory cortex region when correctly modelling shared peptides.

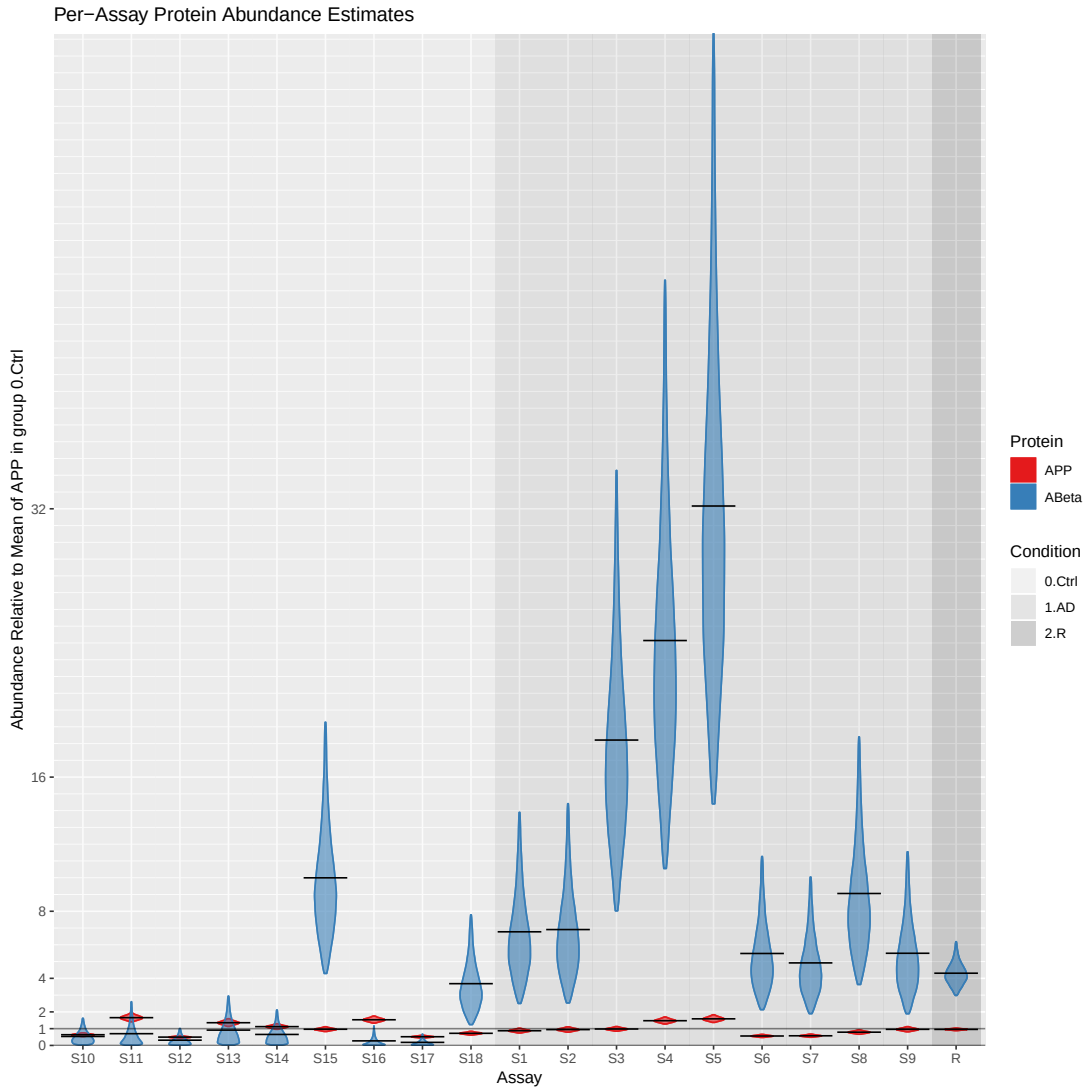

Figure 6.32: Violin plots showing the posterior estimates of the protein-level abundances of the APP and Abeta proteins in the sensory cortex region.

6.7.4 Discussion of Amyloid Results

Analysing the data from the Alzheimer's study with a naïve model which fails to take the shared peptide into account is unable to separate the contribution of the Amyloid precursor protein and the Amyloid beta protein to the shared peptide. Additionally, the naïve model is unable to make inferences about the relative abundances of the two proteins.

Applying the Bayesian shared peptide model to the Alzheimer's study data allows for the contributions of the two proteins to the shared peptide to be deconvoluted, with the additional benefit that new inferences can be made about the relative abundances of the two proteins.

It should be noted that the high-degree of uncertainty of the abundance of amyloid beta peptide for the majority of control samples is due to the the low abundance of the amyloid beta peptide. The naïve analysis concludes that the relative fold-change versus the reference sample is relatively certain. However, since the true abundance of Abeta is much lower than that of APP and therefore much more uncertain, it is correct that the model is more uncertain in the relative fold-change for the majority of the control-group samples. Indeed, the shared peptide analysis suggests the likely absence of Abeta in many of the control group samples. The model described above cannot accurately account for the total absence of a protein; the following section discusses a possible approach that would allow for the presence or absence of proteins (or proteoforms) to be determined.

6.8 Discussion

It was noted in Section 2.2 that many previous attempts to handle shared peptides failed to take into account the variability in ionisation of peptide features. The model described in this chapter explicitly models these ionisation effects and in doing so is able to make inferences about the relative abundance of proteoforms via any peptides that are shared between them.

The relative quantification of otherwise unrelated proteins by creating an artificial bridge proteoform using designer peptides presents a promising application: relative between-proteoform quantification is already possible through spiked-in standard peptides, but this analysis is limited by the accuracy with which the concentration of spike-in peptides can be prepared. Determining the relative between-proteoform concentration using a bridge proteoform would not be predicated on the accuracy of the

bridge proteoform's concentration when spiked-in, only require that it is spiked-in at two different levels so that the ionisation coefficients, and therefore the relative quantification, can be inferred.

The usefulness of this relative abundance estimation is limited where there is a very large difference in abundance between the proteoforms, such as the results presented in Section 6.7, where the apparent low abundance of the amyloid beta peptide in the majority of the control group samples means that the abundance relative to the amyloid precursor protein is particularly uncertain in these samples. This is because the contribution of amyloid beta to the shared peptide is likely to be very small and therefore more uncertain. This also points to a limitation of the model: since protein abundances are modelled as log-normal, where a proteoform or subset protein is observed only through shared peptide(s) the model has no way by which it can conclude that particular proteoform is actually absent. Section 6.9.1 discusses one approach to address this.

The model exhibits some problems with non-identifiability on simulated datasets with small amounts of data, and therefore exhibits awkward posterior geometry which MCMC samplers can struggle to sample from effectively. However, these same problems with identifiability are not apparent when the model is applied to datasets from real experiments, both artificial and clinical, where there are more peptides observed and quantified.

6.9 Conclusions

The potential utility of a hierarchical Bayesian model to the analysis of shared peptides in mass spectrometry proteomics has been demonstrated. By estimating the relative ionisation of each feature, this novel model allows for the estimation of the relative quantification between proteins, including proteoforms and subset proteins. Additionally, the potential for the model to be used in combination with a synthetic biology technology to perform relative absolute quantification of otherwise unrelated proteins has been demonstrated.

6.9.1 Future Work

There is some potential for future development of this shared peptide model.

Further Investigation with Simulated Data

With examples of small, simulated data sets the shared peptide model fails to correctly infer the relative quantification between proteoforms, exhibiting some bias. Inspection of the posterior samples suggest that there is some non-identifiability in the model in these cases. The same issues are not present when analysing real data. This should be investigated further; it is possible that a reparameterisation of the model would solve this unidentifiability and lead to improved inference. Alternatively, a more advanced MCMC sampler using a numerical integrator with an adaptive step-size or an SMC sampler may provide a solution to this problem.

Computational Speedup

As noted in Section 6.3.3, during the development of the model it was noticed that in some cases, a number of MCMC chains would fail to converge to the same region of parameter space; chains adapting to regions of parameter space far from the typical set results in poor mixing. This was mitigated by applying tighter priors and constraining the initialisation of MCMC chains. At present, there is no information shared between chains, hence warm-up, adaptation and convergence to the typical set must be achieved by each MCMC chain independently. More advanced algorithms such as SMC samplers, where the parallelism of chains (or particles) results in information being shared across the population of chains, would allow for better convergence and subsequent sampling from high-dimensional posteriors with awkward geometry.

Variable Selection

The low abundance of the Amyloid beta protein in many of the control samples of the Alzheimer's study hints at a potential improvement to this model: the model described above makes the assumption that all proteins or proteoforms being analysed are present in every sample, even where those proteins or proteoforms are quantified entirely by shared peptides. However it may not be the case that every protein or proteoform is present in every sample. A potentially more accurate model would include variable selection using binary indicator variables on the presence or absence of each proteoform whose presence cannot be discerned before quantification.

Chapter 7

Conclusions and Recommendations

7.1 Conclusions

This thesis aimed to explore some of the problems associated with the quantification of proteins in mass spectrometry proteomics.

It has been demonstrated that the propagation of uncertainty through an analysis pipeline — from LC-MS features, through peptides and proteins, up to differential expression — is crucial to achieve maximum sensitivity.

Marginalisation of parameters through the use of conjugate distributions was demonstrated to improve the efficiency of an MCMC sampler. This simple technique can be applied to a wide class of problems where conjugate distributions can be exploited.

The use of conjugate distributions was then taken further; an approach to differential expression analysis based on a Bayesian model comparison paradigm was proposed, exploiting the analytic structure of the models to achieve fast inference and calibrated false discovery rates. Furthermore, the calibration of false discovery rate estimates was demonstrated to be dependent on both the propagation of uncertainty and for statistical testing to be performed in a global context.

The use of a Bayesian hierarchical model has enabled shared peptides to be included in the protein quantification pipeline. Modelling of the underlying abundances of proteoforms has the added benefit of providing relative quantification between proteoforms without the need for internal standards for the first time.

More generally, the success of Bayesian inference on protein quantification problems

has been shown to be dependent on two main factors: the propagation of uncertainty between analysis steps and a focus on making inference in a global context.

7.2 Recommendations

Building on the validation in this thesis, further validation of the BayesProt model, with a greater variety of input data types, is planned for a forthcoming publication.

Following this, we will extend BayesProt with the Bayesian model comparison in a further publication. Future efforts concerned with improved estimation of false discovery rates should ensure that full population-level context is used to inform decision making.

Future research should seek to extend the quantification of proteoforms to making probabilistic predictions of the presence or absence of all possible proteoforms on a per sample, or per condition basis. Similarly, the uncertainty of protein and peptide identifications could be considered for integration into the quantification process. Analysis of all proteins at once, tying together the joint themes of uncertainty propagation and global context, should be a target for future research. Furthermore, there are other downstream analysis techniques, such as biological pathway analysis, that would benefit from propagation of the quantification uncertainty.

As initiatives to identify biomarkers based on experiments consisting of hundreds of samples continue, the ability of analysis to scale to these large experiments will become increasingly important; robust methods are needed to account for measurement drift where experiments can require multiple weeks to be analysed by mass spectrometry.

The analysis of all proteins at once — especially if combined with larger data sets and the modelling of shared peptides — will result in a high-dimensional posterior, which poses a problem to even the most advanced MCMC samplers. Hence, it is likely that more advanced sampling algorithms running on distributed systems will need to be employed to tackle these models for proteome-wide inference.

Appendix A

Probability Distributions

Below is a brief description of some of the probability distributions used in this thesis for clarity and completeness, since several different parameterisations are in common usage for some.

A.1 Normal Distribution

The probability density function takes the form:

$$P\left(x|\mu,\sigma\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{A.1}$$

A.2 Log-normal Distribution

The probability density function takes the form:

$$P\left(x|\mu,\sigma\right) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\log\left(x\right)-\mu}{\sigma}\right)^2} \tag{A.2}$$

A.3 Gamma Distribution

We use the shape-rate parameterisation of the gamma distribution, with shape parameter α and rate parameter β. The probability density function takes the form:

$$P\left(x|\alpha,\beta\right) = \frac{\beta^{\alpha} x^{\alpha-1} e^{-\beta x}}{\Gamma\left(\alpha\right)} \tag{A.3}$$

A.4 Poisson Distribution

The Poisson distribution describes the probability of a number of event k occurring in a fixed interval with mean rate λ. The probability density function takes the form:

$$P\left(k|\lambda\right) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{A.4}$$

A.5 Negative Binomial Distribution

The negative binomial distribution describes the number of successes k before r failures of successive Bernoulli trials with probability of success, p. The probability density function is of the form:

$$P\left(k|r,p\right) = \binom{k+r-1}{k} \cdot (1-p)^r \cdot p^k \tag{A.5}$$

A.6 Scale-Inverse-Chi-Squared Distribution

The probability density function takes the form:

$$P\left(x|\mu,\tau^2\right) = \frac{(\tau^2\nu/2)^{\nu/2}}{\Gamma(\nu/2)} \frac{e^{\frac{-\nu\tau^2}{2x}}}{x^{1+\nu/2}} \tag{A.6}$$

A.7 Multivariate Normal Distribution

The probability density function takes the form:

$$P\left(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}\right) = (2\pi)^{-k/2}|\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \tag{A.7}$$

where $\mathbf{x}$ is d-dimensional, $\boldsymbol{\mu}$ is the d-dimensional location parameter and $\boldsymbol{\Sigma}$ is a $d \times d$ covariance matrix.

A.8 Multivariate Student-T Distribution

The probability density function takes the form:

$$P\left(\mathbf{x}|\nu,\boldsymbol{\mu},\boldsymbol{\Sigma},\right)=\frac{\Gamma\left((\nu+d)/2\right)}{\Gamma\left(\nu/2\right)\nu^{d/2}\pi^{d/2}}|\boldsymbol{\Sigma}|^{-1/2}\left[1+\frac{1}{\nu}\left(\mathbf{x}-\boldsymbol{\mu}\right)^{T}\boldsymbol{\Sigma}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}\right)\right]^{-(\nu+d)/2} \tag{A.8}$$

where $\mathbf{x}$ is d-dimensional, ν denotes the degrees of freedom, $\boldsymbol{\mu}$ is the d-dimensional location parameter and $\boldsymbol{\Sigma}$ is a $d \times d$ scaling matrix.

A.9 Multivariate Normal-Scale-Inverse-Chi-Squared Distribution

The probability density function takes the form:

$$P\left(\mathbf{x},\sigma^{2}|\boldsymbol{\mu},\mathbf{V},\nu,\tau^{2}\right)=\frac{\left(\nu\tau^{2}/2\right)^{\nu/2}}{\left(2\pi\right)^{d/2}\left|\mathbf{V}\right|^{1/2}\Gamma(\nu/2)}\frac{\exp\left(-[(\mathbf{x}-\boldsymbol{\mu})^{T}\mathbf{V}^{-1}(\mathbf{x}-\boldsymbol{\mu})+\nu\tau^{2}]/(2\sigma^{2})\right)}{\left(\sigma^{2}\right)^{-(\nu+d+2)/2}} \tag{A.9}$$

where $\mathbf{x}$ is d-dimensional, $\boldsymbol{\mu}$ is the d-dimensional location parameter, $\mathbf{V}$ is a $d \times d$ covariance matrix, ν denotes the degrees of freedom and τ^2 is a scaling parameter.

Appendix B

Updating Multivariate Normal-Scaled-Inverse-Chi-Squared Prior Parameters

The parameters $\boldsymbol{\mu}_0$, $\mathbf{V}$, ν, τ^2 of the normal-scaled-inverse-χ^2 prior on $\boldsymbol{\mu}$ and σ can be updated conditional on observed data $\mathbf{y}$ and condition matrix $\mathbf{X}$ to give updated parameters $\boldsymbol{\mu}_0'$, $\mathbf{V}'$, ν', τ'^2:

$$\boldsymbol{\mu}_0' = (\mathbf{V}^{-1} + \mathbf{X}^\intercal\mathbf{X})^{-1}(\mathbf{V}^{-1}\boldsymbol{\mu}_0 + \mathbf{X}^\intercal\mathbf{y}^\intercal) \tag{B.1}$$

$$\mathbf{V}' = (\mathbf{V}^{-1} + \mathbf{X}^\intercal\mathbf{X})^{-1} \tag{B.2}$$

$$\nu' = \nu + n \tag{B.3}$$

$$\tau'^2 = \frac{\nu\tau^2}{\nu'} + \frac{1}{\nu'}(\boldsymbol{\mu}_0^\intercal\mathbf{V}^{-1}\boldsymbol{\mu}_0 + \mathbf{y}^\intercal\mathbf{y} - (\boldsymbol{\mu}_0')^\intercal(\mathbf{V}')^{-1}\boldsymbol{\mu}_0') \tag{B.4}$$

A full derivation of these equations for the equivalent multivariate-normal-inverse-gamma can be found in [128].

Appendix C

Derivation of Jacobian of Feature Transform

C.1 1-Simplex Case

$x = f(\beta)$

$$[\beta] \xmapsto{f} [\frac{\beta}{1+\beta}] \tag{C.1}$$

Jacobian:

$$\left|\frac{\partial x}{\partial \beta}\right| = \left|\frac{\partial(\frac{\beta}{\beta+1})}{\partial \beta}\right| \tag{C.2}$$

$$= \left|\frac{\frac{\partial(\beta)}{\partial \beta}\cdot(1+\beta) - \frac{\partial(\beta+1)}{\partial \beta}\cdot\beta}{(1+\beta)^2}\right| \tag{C.3}$$

$$= \left|\frac{1\cdot(\beta+1) - 1\cdot\beta}{(1+\beta)^2}\right| \tag{C.4}$$

$$= (\beta+1)^{-2} \tag{C.5}$$

C.2 K-1 Simplex Case

$x_{1:K-1} = f(\beta_{1:K-1})$

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{K-1} \end{bmatrix} \overset{f}{\longmapsto} \begin{bmatrix} \frac{\beta_1}{1+\sum_{i=1}^{K-1}\beta_i} \\ \frac{\beta_2}{1+\sum_{i=1}^{K-1}\beta_i} \\ \vdots \\ \frac{\beta_{K-1}}{1+\sum_{i=1}^{K-1}\beta_i} \end{bmatrix} \tag{C.6}$$

Jacobian:

$$\left|J\right| = \begin{vmatrix} \frac{\partial x_1}{\partial \beta_1} & \frac{\partial x_1}{\partial \beta_2} & \cdots & \frac{\partial x_1}{\partial \beta_{K-1}} \\ \frac{\partial x_2}{\partial \beta_1} & \frac{\partial x_2}{\partial \beta_2} & \cdots & \frac{\partial x_2}{\partial \beta_{K-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_{K-1}}{\partial \beta_1} & \frac{\partial x_{K-1}}{\partial \beta_2} & \cdots & \frac{\partial x_{K-1}}{\partial \beta_{K-1}} \end{vmatrix} \tag{C.7}$$

$$= \begin{vmatrix} \frac{(1+\sum\beta_i)-\beta_1}{(1+\sum\beta_i)^2} & \frac{-\beta_1}{(1+\sum\beta_i)^2} & \cdots & \frac{-\beta_1}{(1+\sum\beta_i)^2} \\ \frac{-\beta_2}{(1+\sum\beta_i)^2} & \frac{(1+\sum\beta_i)-\beta_2}{(1+\sum\beta_i)^2} & \cdots & \frac{-\beta_2}{(1+\sum\beta_i)^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{-\beta_{K-1}}{(1+\sum\beta_i)^2} & \frac{-\beta_{K-1}}{(1+\sum\beta_i)^2} & \cdots & \frac{(1+\sum\beta_i)-\beta_{K-1}}{(1+\sum\beta_i)^2} \end{vmatrix} \tag{C.8}$$

Then, using the fact that $det(cA) = c^n det(A)$ for an nxn matrix:

$$= \begin{vmatrix} (1+\sum\beta_i)-\beta_1 & -\beta_1 & \cdots & -\beta_1 \\ -\beta_2 & (1+\sum\beta_i)-\beta_2 & \cdots & -\beta_2 \\ \vdots & \vdots & \ddots & \vdots \\ -\beta_{K-1} & -\beta_{K-1} & \cdots & (1+\sum\beta_i)-\beta_{K-1} \end{vmatrix} \cdot \left(\frac{1}{(1+\sum\beta_i)^2}\right)^{K-1} \tag{C.9}$$

$$= \begin{vmatrix} (1+\sum\beta_i)-\beta_1 & -\beta_1 & \cdots & -\beta_1 \\ -\beta_2 & (1+\sum\beta_i)-\beta_2 & \cdots & -\beta_2 \\ \vdots & \vdots & \ddots & \vdots \\ -\beta_{K-1} & -\beta_{K-1} & \cdots & (1+\sum\beta_i)-\beta_{K-1} \end{vmatrix} \cdot \left(1+\sum\beta_i\right)^{-2K+2} \tag{C.10}$$

Then, adding every other row to the first (which does not affect the value of the determinant):

$$= \begin{vmatrix} 1 & 1 & \cdots & 1 \\ -\beta_2 & (1+\sum\beta_i)-\beta_2 & \cdots & -\beta_2 \\ \vdots & \vdots & \ddots & \vdots \\ -\beta_{K-1} & -\beta_{K-1} & \cdots & (1+\sum\beta_i)-\beta_{K-1} \end{vmatrix} \cdot \left(1+\sum\beta_i\right)^{-2K+2} \tag{C.11}$$

Subtracting the first column from each of the other columns (which again does not affect the value of the determinant):

$$= \begin{vmatrix} 1 & 0 & \cdots & 0 \\ -\beta_2 & (1+\sum\beta_i) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -\beta_{K-1} & 0 & \cdots & (1+\sum\beta_i) \end{vmatrix} \cdot \left(1+\sum\beta_i\right)^{-2K+2} \tag{C.12}$$

$$= 1 \cdot \begin{vmatrix} (1+\sum\beta_i) & 0 & \cdots & 0 \\ 0 & (1+\sum\beta_i) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (1+\sum\beta_i) \end{vmatrix} \cdot \left(1+\sum\beta_i\right)^{-2K+2} \tag{C.13}$$

$$= \left(1+\sum\beta_i\right)^{K-2} \cdot \left(1+\sum\beta_i\right)^{-2K+2} \tag{C.14}$$

$$= \left(1+\sum\beta_i\right)^{-K} \tag{C.15}$$

Bibliography

[1] O. J. Freeman, R. D. Unwin, A. W. Dowsey, P. Begley, S. Ali, K. A. Hollywood, N. Rustogi, R. S. Petersen, W. B. Dunn, G. J. Cooper, and N. J. Gardiner, "Metabolic dysfunction is restricted to the sciatic nerve in experimental diabetic neuropathy," *Diabetes*, db150835, Oct. 15, 2015.

[2] J. Xu, S. Patassini, N. Rustogi, I. Riba-Garcia, B. D. Hale, A. M. Phillips, H. Waldvogel, R. Haines, P. Bradbury, A. Stevens, R. L. M. Faull, A. W. Dowsey, G. J. S. Cooper, and R. D. Unwin, "Regional protein expression in human Alzheimer's brain correlates with disease severity," *Communications Biology*, vol. 2, no. 1, p. 43, Feb. 4, 2019.

[3] H. Liao, A. Phillips, A. Jankevics, and A. W. Dowsey, "Chapter 7 Algorithms for MS1-Based Quantitation," in *Proteome Informatics*, The Royal Society of Chemistry, 2017, pp. 133–154.

[4] M. Bantscheff, S. Lemeer, M. M. Savitski, and B. Kuster, "Quantitative mass spectrometry in proteomics: Critical review update from 2007 to the present," *Anal Bioanal Chem*, vol. 404, no. 4, pp. 939–965, Sep. 1, 2012.

[5] O. T. Schubert, H. L. Röst, B. C. Collins, G. Rosenberger, and R. Aebersold, "Quantitative proteomics: Challenges and opportunities in basic and applied research," *Nature Protocols*, vol. 12, no. 7, pp. 1289–1294, Jul. 2017.

[6] G. Cagney, S. Amiri, T. Premawaradena, M. Lindo, and A. Emili, "In silico proteome analysis to facilitate proteomics experiments using mass spectrometry," *Proteome Science*, vol. 1, no. 1, p. 5, Aug. 13, 2003.

[7] L. C. Gillet, P. Navarro, S. Tate, H. Röst, N. Selevsek, L. Reiter, R. Bonner, and R. Aebersold, "Targeted Data Extraction of the MS/MS Spectra Generated by Data-independent Acquisition: A New Concept for Consistent and Accurate Proteome Analysis," *Molecular & Cellular Proteomics*, vol. 11, no. 6, O111.016717, Jun. 1, 2012.

[8] D. N. Perkins, D. J. C. Pappin, D. M. Creasy, and J. S. Cottrell, "Probability-based protein identification by searching sequence databases using mass spectrometry data," *Electrophoresis*, vol. 20, no. 18, pp. 3551–3567, Dec. 1, 1999.

[9] L. C. Gillet, A. Leitner, and R. Aebersold, "Mass Spectrometry Applied to Bottom-Up Proteomics: Entering the High-Throughput Era for Hypothesis Testing," *Annual Review of Analytical Chemistry*, vol. 9, no. 1, pp. 449–472, 2016.

[10] P. Brownridge and R. J. Beynon, "The importance of the digest: Proteolysis and absolute quantification in proteomics," *Methods*, vol. 54, no. 4, pp. 351–360, Aug. 2011.

[11] W. M. Old, K. Meyer-Arendt, L. Aveline-Wolf, K. G. Pierce, A. Mendoza, J. R. Sevinsky, K. A. Resing, and N. G. Ahn, "Comparison of label-free methods for quantifying human proteins by shotgun proteomics," *Molecular & Cellular Proteomics*, vol. 4, no. 10, pp. 1487–1502, Oct. 2005.

[12] S.-E. Ong, B. Blagoev, I. Kratchmarova, D. B. Kristensen, H. Steen, A. Pandey, and M. Mann, "Stable Isotope Labeling by Amino Acids in Cell Culture, SILAC, as a Simple and Accurate Approach to Expression Proteomics," *Molecular & Cellular Proteomics*, vol. 1, no. 5, pp. 376–386, May 2002.

[13] S. Wiese, K. A. Reidegeld, H. E. Meyer, and B. Warscheid, "Protein labeling by iTRAQ: A new tool for quantitative mass spectrometry in proteome research," *PROTEOMICS*, vol. 7, no. 3, pp. 340–350, Feb. 2007.

[14] A. Thompson, J. Schäfer, K. Kuhn, S. Kienle, J. Schwarz, G. Schmidt, T. Neumann, and C. Hamon, "Tandem Mass Tags: A Novel Quantification Strategy for Comparative Analysis of Complex Protein Mixtures by MS/MS," *Analytical Chemistry*, vol. 75, no. 8, pp. 1895–1904, Apr. 2003.

[15] T. Werner, I. Becher, G. Sweetman, C. Doce, M. M. Savitski, and M. Bantscheff, "High-Resolution Enabled TMT 8-plexing," *Analytical Chemistry*, vol. 84, no. 16, pp. 7188–7194, Aug. 21, 2012.

[16] V. Vidova and Z. Spacil, "A review on mass spectrometry-based quantitative proteomics: Targeted and data independent acquisition," *Analytica Chimica Acta*, vol. 964, pp. 7–23, Apr. 29, 2017.

[17] C. Ludwig, L. Gillet, G. Rosenberger, S. Amon, B. C. Collins, and R. Aebersold, "Data-independent acquisition-based SWATH-MS for quantitative proteomics: A tutorial," *Molecular Systems Biology*, vol. 14, no. 8, e8126, Aug. 1, 2018.

[18] 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.

[19] B. Carrillo, C. Yanofsky, S. Laboissiere, R. Nadon, and R. E. Kearney, "Methods for combining peptide intensities to estimate relative protein abundance," *Bioinformatics*, vol. 26, no. 1, pp. 98–103, Jan. 1, 2010.

[20] R. Smith, D. Ventura, and J. T. Prince, "LC-MS alignment in theory and practice: A comprehensive algorithmic review," *Brief Bioinform*, vol. 16, no. 1, pp. 104–117, Jan. 1, 2015.

[21] T. Välikangas, T. Suomi, and L. L. Elo, "A systematic evaluation of normalization methods in quantitative label-free proteomics," *Briefings in Bioinformatics*, bbw095, Oct. 2, 2016.

[22] B. Bolstad, R. Irizarry, M. Astrand, and T. Speed, "A comparison of normalization methods for high density oligonucleotide array data based on variance and bias," *Bioinformatics*, vol. 19, no. 2, pp. 185–193, Jan. 22, 2003.

[23] C. Ammar, M. Gruber, G. Csaba, and R. Zimmer, "MS-Empire utilizes peptide-level noise distributions for ultra sensitive detection of differentially expressed proteins," *Molecular & Cellular Proteomics*, mcp.RA119.001509, Jan. 1, 2019.

[24] J. Cox, M. Y. Hein, C. A. Luber, I. Paron, N. Nagaraj, and M. Mann, "Accurate Proteome-wide Label-free Quantification by Delayed Normalization and Maximal Peptide Ratio Extraction, Termed MaxLFQ," *Mol Cell Proteomics*, vol. 13, no. 9, pp. 2513–2526, Jan. 9, 2014.

[25] Y. V. Karpievitch, T. Taverner, J. N. Adkins, S. J. Callister, G. A. Anderson, R. D. Smith, and A. R. Dabney, "Normalization of peak intensities in bottom-up MS-based proteomics using singular value decomposition," *Bioinformatics*, vol. 25, no. 19, pp. 2573–2580, Oct. 1, 2009.

[26] O. Serang and W. Noble, "A review of statistical methods for protein identification using tandem mass spectrometry," *Stat Interface*, vol. 5, no. 1, pp. 3–20, 2012.

[27] B. Dost, N. Bandeira, X. Li, Z. Shen, S. P. Briggs, and V. Bafna, "Accurate Mass Spectrometry Based Protein Quantification via Shared Peptides," *Journal of Computational Biology*, vol. 19, no. 4, pp. 337–348, Apr. 2012.

[28] 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, no. 12, pp. 1367–1372, Dec. 2008.

[29] A. L. Oberg and O. Vitek, "Statistical Design of Quantitative Mass Spectrometry-Based Proteomic Experiments," *Journal of Proteome Research*, vol. 8, no. 5, pp. 2144–2156, May 2009.

[30] W. Huber, A. von Heydebreck, H. Sultmann, A. Poustka, and M. Vingron, "Variance stabilization applied to microarray data calibration and to the quantification of differential expression," *Bioinformatics*, vol. 18, S96–S104, Suppl 1 Jul. 1, 2002.

[31] L. J. E. Goeminne, K. Gevaert, and L. Clement, "Peptide-level Robust Ridge Regression Improves Estimation, Sensitivity, and Specificity in Data-dependent Quantitative Label-free Shotgun Proteomics," *Molecular & Cellular Proteomics*, vol. 15, no. 2, pp. 657–668, Feb. 2016.

[32] M. Choi, C.-Y. Chang, T. Clough, D. Broudy, T. Killeen, B. MacLean, and O. Vitek, "MSstats: An R package for statistical analysis of quantitative mass spectrometry-based proteomic experiments," *Bioinformatics*, vol. 30, no. 17, pp. 2524–2526, Sep. 1, 2014.

[33] T. Clough, S. Thaminy, S. Ragg, R. Aebersold, and O. Vitek, "Statistical protein quantification and significance analysis in label-free LC-MS experiments with complex designs," *BMC Bioinformatics*, vol. 13, S6, Suppl 16 2012.

[34] Y. Karpievitch, J. Stanley, T. Taverner, J. Huang, J. N. Adkins, C. Ansong, F. Heffron, T. O. Metz, W.-J. Qian, H. Yoon, R. D. Smith, and A. R. Dabney, "A statistical framework for protein quantitation in bottom-up MS-based proteomics," *Bioinformatics*, vol. 25, no. 16, pp. 2028–2034, Aug. 15, 2009.

[35] T. Clough, M. Key, I. Ott, S. Ragg, G. Schadow, and O. Vitek, "Protein Quantification in Label-Free LC-MS Experiments," *Journal of Proteome Research*, vol. 8, no. 11, pp. 5275–5284, Nov. 2009.

[36] A. L. Oberg, D. W. Mahoney, J. E. Eckel-Passow, C. J. Malone, R. D. Wolfinger, E. G. Hill, L. T. Cooper, O. K. Onuma, C. Spiro, T. M. Therneau, and H. R. Bergen III, "Statistical Analysis of Relative Labeled Mass Spectrometry Data from Complex Samples Using ANOVA," *Journal of Proteome Research*, vol. 7, no. 1, pp. 225–233, Jan. 2008.

[37] E. G. Hill, J. H. Schwacke, S. Comte-Walters, E. H. Slate, A. L. Oberg, J. E. Eckel-Passow, T. M. Therneau, and K. L. Schey, "A Statistical Model for iTRAQ Data Analysis," *Journal of Proteome Research*, vol. 7, no. 8, pp. 3091–3101, Aug. 2008.

[38] H. Jow, R. J. Boys, and D. J. Wilkinson, "Bayesian identification of protein differential expression in multi-group isobaric labelled mass spectrometry data," *Statistical Applications in Genetics and Molecular Biology*, vol. 13, no. 5, Jan. 1, 2014.

[39] A. Gelman, J. Hill, and M. Yajima, "Why We (Usually) Don't Have to Worry About Multiple Comparisons," *Journal of Research on Educational Effectiveness*, vol. 5, no. 2, pp. 189–211, Apr. 2012.

[40] G. K. Smyth, "Linear models and empirical bayes methods for assessing differential expression in microarray experiments," *Stat Appl Genet Mol Biol*, vol. 3, Article3, 2004.

[41] A. Sticker, L. Goeminne, L. Martens, and L. Clement, "Robust summarization and inference in proteome-wide label-free quantification," *bioRxiv*, p. 668 863, Jun. 13, 2019.

[42] M. The and L. Käll, "Integrated Identification and Quantification Error Probabilities for Shotgun Proteomics," *Molecular & Cellular Proteomics*, vol. 18, no. 3, pp. 561–570, Mar. 1, 2019.

[43] H. Choi, S. Kim, D. Fermin, C.-C. Tsou, and A. I. Nesvizhskii, "QPROT: Statistical method for testing differential expression using protein-level intensity data in label-free quantitative proteomics," *Journal of Proteomics*, vol. 129, pp. 121–126, Nov. 2015.

[44] G. K. Smyth, "Limma: Linear models for microarray data," in *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, Springer, 2005, pp. 397–420.

[45] K. Kammers, R. N. Cole, C. Tiengwe, and I. Ruczinski, "Detecting significant changes in protein abundance," *EuPA Open Proteomics*, vol. 7, pp. 11–19, Jun. 2015.

[46] L. M. Smith, N. L. Kelleher, The Consortium for Top Down Proteomics, M. Linial, D. Goodlett, P. Langridge-Smith, Y. Ah Goo, G. Safford, L. Bonilla, G. Kruppa, R. Zubarev, J. Rontree, J. Chamot-Rooke, J. Garavelli, A. Heck, J. Loo, D. Penque, M. Hornshaw, C. Hendrickson, L. Pasa-Tolic, C. Borchers,

D. Chan, N. Young, J. Agar, C. Masselon, M. Gross, F. McLafferty, Y. Tsybin, Y. Ge, I. Sanders, J. Langridge, J. Whitelegge, and A. Marshall, "Proteoform: A single term describing protein complexity," *Nature Methods*, vol. 10, no. 3, pp. 186–187, Mar. 2013.

[47] K. Podwojski, M. Eisenacher, M. Kohl, M. Turewicz, H. E. Meyer, J. Rahnenführer, and C. Stephan, "Peek a peak: A glance at statistics for quantitative label-free proteomics," *Expert Review of Proteomics*, vol. 7, no. 2, pp. 249–261, Apr. 2010.

[48] M. Blein-Nicolas, H. Xu, D. de Vienne, C. Giraud, S. Huet, and M. Zivy, "Including shared peptides for estimating protein abundances: A significant improvement for quantitative proteomics," *Proteomics*, vol. 12, no. 18, pp. 2797–2801, Sep. 2012.

[49] B.-J. M. Webb-Robertson, M. M. Matzke, S. Datta, S. H. Payne, J. Kang, L. M. Bramer, C. D. Nicora, A. K. Shukla, T. O. Metz, K. D. Rodland, R. D. Smith, M. F. Tardiff, J. E. McDermott, J. G. Pounds, and K. M. Waters, "Bayesian Proteoform Modeling Improves Protein Quantification of Global Proteomic Measurements," *Molecular & Cellular Proteomics*, vol. 13, no. 12, pp. 3639–3646, Dec. 2014.

[50] S. Jin, D. S. Daly, D. L. Springer, and J. H. Miller, "The Effects of Shared Peptides on Protein Quantitation in Label-Free Proteomics by LC/MS/MS," *Journal of Proteome Research*, vol. 7, no. 1, pp. 164–169, Jan. 2008.

[51] Y. Zhang, Z. Wen, M. P. Washburn, and L. Florens, "Refinements to Label Free Proteome Quantitation: How to Deal with Peptides Shared by Multiple Proteins," *Analytical Chemistry*, vol. 82, no. 6, pp. 2272–2281, Mar. 15, 2010.

[52] ——, "Improving Label-Free Quantitative Proteomics Strategies by Distributing Shared Peptides and Stabilizing Variance," *Analytical Chemistry*, vol. 87, no. 9, pp. 4749–4756, May 5, 2015.

[53] S. Gerster, T. Kwon, C. Ludwig, M. Matondo, C. Vogel, E. M. Marcotte, R. Aebersold, and P. Buehlmann, "Statistical Approach to Protein Quantification," *Molecular & Cellular Proteomics*, vol. 13, no. 2, pp. 666–677, Feb. 2014.

[54] L. Jacob, F. Combes, and T. Burger, "PEPA test: Fast and powerful differential analysis from relative quantitative proteomics data using shared peptides," *Biostatistics*, Jun. 18, 2018.

[55] B. He, J. Shi, X. Wang, H. Jiang, and H.-J. Zhu, "Label-free absolute protein quantification with data-independent acquisition," *Journal of Proteomics*, vol. 200, pp. 51–59, May 30, 2019.

[56] A. Pursiheimo, A. P. Vehmas, S. Afzal, T. Suomi, T. Chand, L. Strauss, M. Poutanen, A. Rokka, G. L. Corthals, and L. L. Elo, "Optimization of Statistical Methods Impact on Quantitative Proteomics Data.," *Journal of proteome research*, vol. 14, no. 10, pp. 4118–4126, 2015 Oct 2 (Epub 2015 Sep 08).

[57] B. Phipson, S. Lee, I. J. Majewski, W. S. Alexander, and G. K. Smyth, "Robust hyperparameter estimation protects against hypervariable genes and improves power to detect differential expression," *Ann Appl Stat*, vol. 10, no. 2, pp. 946–963, Jun. 2016.

[58] H. Choi and A. I. Nesvizhskii, "False Discovery Rates and Related Statistical Concepts in Mass Spectrometry-Based Proteomics," *J. Proteome Res.*, vol. 7, no. 1, pp. 47–50, Jan. 1, 2008.

[59] T. Burger, "Gentle Introduction to the Statistical Foundations of False Discovery Rate in Quantitative Proteomics," *Journal of Proteome Research*, vol. 17, no. 1, pp. 12–22, Jan. 5, 2018.

[60] Y. Benjamini and Y. Hochberg, "Controlling the False Discovery Rate - a Practical and Powerful Approach to Multiple Testing," *Journal of the Royal Statistical Society Series B-Methodological*, vol. 57, no. 1, pp. 289–300, 1995.

[61] K. Korthauer, P. K. Kimes, C. Duvallet, A. Reyes, A. Subramanian, M. Teng, C. Shukla, E. J. Alm, and S. C. Hicks, "A practical guide to methods controlling false discoveries in computational biology," *Genome Biology*, vol. 20, no. 1, p. 118, Jun. 4, 2019.

[62] Y. Ge, S. C. Sealfon, and T. P. Speed, "Multiple testing and its applications to microarrays," *Stat Methods Med Res*, vol. 18, no. 6, pp. 543–563, Dec. 2009.

[63] W. S. Noble and M. J. MacCoss, "Computational and Statistical Analysis of Protein Mass Spectrometry Data," *PLoS Computational Biology*, vol. 8, no. 1, P. E. Bourne, Ed., e1002296, Jan. 26, 2012.

[64] D. Pascovici, D. C. L. Handler, J. X. Wu, and P. A. Haynes, "Multiple testing corrections in quantitative proteomics: A useful but blunt tool," *PROTEOMICS*, vol. 16, no. 18, pp. 2448–2453, 2016.

[65] J. D. Storey, "The positive false discovery rate: A Bayesian interpretation and the q -value," *The Annals of Statistics*, vol. 31, no. 6, pp. 2013–2035, Dec. 2003.

[66] N. Ignatiadis, B. Klaus, J. B. Zaugg, and W. Huber, "Data-driven hypothesis weighting increases detection power in genome-scale multiple testing," *Nature Methods*, vol. 13, no. 7, pp. 577–580, Jul. 2016.

[67] S. M. Boca and J. T. Leek, "A direct approach to estimating false discovery rates conditional on covariates," *PeerJ*, vol. 6, e6035, Dec. 10, 2018.

[68] A. Lewin, N. Bochkina, and S. Richardson, "Fully Bayesian Mixture Model for Differential Gene Expression: Simulations and Model Checks," *Statistical Applications in Genetics and Molecular Biology*, vol. 6, no. 1, Jan. 21, 2007.

[69] J. S. Morris, P. J. Brown, R. C. Herrick, K. A. Baggerly, and K. R. Coombes, "Bayesian Analysis of Mass Spectrometry Proteomic Data Using Wavelet-Based Functional Mixed Models," *Biometrics*, vol. 64, no. 2, pp. 479–489, Jun. 2008.

[70] M. Stephens, "False discovery rates: A new deal," *Biostatistics*, vol. 18, no. 2, pp. 275–294, Apr. 2017.

[71] A. Gelman, *Bayesian Data Analysis, Third Edition*. Chapman and Hall/CRC, Nov. 27, 2013.

[72] J. K. Kruschke, *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan*, Edition 2. Boston: Academic Press, 2015, 759 pp.

[73] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid Monte Carlo," *Physics Letters B*, vol. 195, no. 2, pp. 216–222, Sep. 3, 1987.

[74] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, Jun. 1, 1953.

[75] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, Apr. 1, 1970.

[76] G. O. Roberts, A. Gelman, and W. R. Gilks, "Weak convergence and optimal scaling of random walk Metropolis algorithms," *Ann. Appl. Probab.*, vol. 7, no. 1, pp. 110–120, Feb. 1997.

[77] H. Haario, E. Saksman, and J. Tamminen, "An Adaptive Metropolis Algorithm," *Bernoulli*, vol. 7, no. 2, p. 223, Apr. 2001.

[78] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.

[79] M. D. Hoffman and A. Gelman, "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo," p. 31, Apr. 2014.

[80] R. M. Neal, "MCMC Using Hamiltonian Dynamics," in *Handbook of Markov Chain Monte Carlo*, vol. 2, 11 vols., May 10, 2011, p. 2.

[81] B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, and A. Riddell, "Stan: A probabilistic programming language," *Journal of Statistical Software*, 2016.

[82] A. Gelman and D. B. Rubin, "Inference from Iterative Simulation Using Multiple Sequences," *Statist. Sci.*, vol. 7, no. 4, pp. 457–472, Nov. 1992.

[83] G. Casella, "An Introduction to Empirical Bayes Data Analysis," *The American Statistician*, vol. 39, no. 2, pp. 83–87, 1985.

[84] P. L. Green and S. Maskell, "Estimating the parameters of dynamical systems from Big Data using Sequential Monte Carlo samplers," *Mechanical Systems and Signal Processing*, vol. 93, pp. 379–396, Sep. 1, 2017.

[85] A. Varsi, L. Kekempanos, J. Thiyagalingam, and S. Maskell, "A Single SMC Sampler on MPI that Outperforms a Single MCMC Sampler," May 24, 2019.

[86] M. A. Newton and A. E. Raftery, "Approximate Bayesian Inference with the Weighted Likelihood Bootstrap," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 56, no. 1, pp. 3–48, 1994.

[87] X.-L. Meng and W. H. Wong, "Simulating Ratios of Normalizing Constants via a Simple Identity: A Theoretical Exploration," p. 30, 1996.

[88] A. Gelman and X.-L. Meng, "Simulating normalizing constants: From importance sampling to bridge sampling to path sampling," *Statistical Science*, vol. 13, no. 2, pp. 163–185, May 1998.

[89] S. Chib, "Marginal Likelihood from the Gibbs Output," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1313–1321, 1995.

[90] S. Chib and I. Jeliazkov, "Marginal Likelihood From the Metropolis–Hastings Output," *Journal of the American Statistical Association*, vol. 96, no. 453, pp. 270–281, Mar. 2001.

[91] J. M. Dickey and B. P. Lientz, "The Weighted Likelihood Ratio, Sharp Hypotheses about Chances, the Order of a Markov Chain," *Ann. Math. Statist.*, vol. 41, no. 1, pp. 214–226, Feb. 1970.

[92] C. P. Robert, D. Wraith, P. M. Goggans, and C.-Y. Chan, "Computational methods for Bayesian model choice," 2009, pp. 251–262.

[93] N. Friel and J. Wyse, "Estimating the evidence – a review," Nov. 8, 2011.

[94] Q. F. Gronau, H. Singmann, and E.-J. Wagenmakers, "Bridgesampling: An R Package for Estimating Normalizing Constants," Oct. 23, 2017.

[95] Q. F. Gronau, A. Sarafoglou, D. Matzke, A. Ly, U. Boehm, M. Marsman, D. S. Leslie, J. J. Forster, E.-J. Wagenmakers, and H. Steingroever, "A Tutorial on Bridge Sampling," Mar. 17, 2017.

[96] A. Mootoovaloo, B. A. Bassett, and M. Kunz, "Bayes Factors via Savage-Dickey Supermodels," Sep. 7, 2016.

[97] T. Äijö, V. Butty, Z. Chen, V. Salo, S. Tripathi, C. B. Burge, R. Lahesmaa, and H. Lähdesmäki, "Methods for time series analysis of RNA-seq data with application to human Th17 cell differentiation," *Bioinformatics*, vol. 30, no. 12, pp. i113–i120, Jun. 15, 2014.

[98] E. Hajiramezanali, S. Z. Dadaneh, P. de Figueiredo, S.-H. Sze, M. Zhou, and X. Qian, "Differential Expression Analysis of Dynamical Sequencing Count Data with a Gamma Markov Chain," Mar. 7, 2018.

[99] P. J. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, no. 4, pp. 711–732, 1995.

[100] P. Papastamoulis and M. Rattray, "A Bayesian model selection approach for identifying differentially expressed transcripts from RNA-Seq data," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 67, no. 1, pp. 3–23, Jan. 2018.

[101] S. Sinharay and H. S. Stern, "An Empirical Comparison of Methods for Computing Bayes Factors in Generalized Linear Mixed Models," *Journal of Computational and Graphical Statistics*, vol. 14, no. 2, 2005.

[102] *Bayesprot - Bayesian Linear Mixed-Effects Model for Protein-Level Quantification and Study-Level Statistical Testing in Proteomics.*

[103] R Core Team, *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2019.

[104] J. D. Hadfield, "MCMC Methods for Multi-Response Generalized Linear Mixed Models: The MCMCglmm R Package," *Journal of Statistical Software*, vol. 33, no. 2, pp. 1–22, 2010.

[105] S. Kassab, P. Begley, S. J. Church, S. M. Rotariu, C. Chevalier-Riffard, A. W. Dowsey, A. M. Phillips, L. A. H. Zeef, B. Grayson, J. C. Neill, G. J. S. Cooper, R. D. Unwin, and N. J. Gardiner, "Cognitive dysfunction in diabetic rats is prevented by pyridoxamine treatment. A multidisciplinary investigation," *Molecular Metabolism*, Aug. 5, 2019.

[106] A. Gelman, "Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper)," *Bayesian Anal.*, vol. 1, no. 3, pp. 515–534, Sep. 2006.

[107] M. L. Delignette-Muller and C. Dutang, "Fitdistrplus: An R Package for Fitting Distributions," *Journal of Statistical Software*, vol. 64, no. 4, pp. 1–34, 2015.

[108] W. Viechtbauer, "Conducting Meta-Analyses in R with the metafor Package," *Journal of Statistical Software*, vol. 36, no. 1, pp. 1–48, Aug. 5, 2010.

[109] J. D. Storey, A. J. Bass, A. Dabney, and D. Robinson, *Qvalue: Q-Value Estimation for False Discovery Rate Control*. 2019.

[110] M. Stephens, P. Carbonetto, D. Gerard, M. Lu, L. Sun, J. Willwerscheid, and N. Xiao, *Ashr: Methods for Adaptive Shrinkage, Using Empirical Bayes*. 2019.

[111] T. Huang, M. Choi, S. Hao, and O. Vitek, *MSstatsTMT: Protein Significance Analysis in Shotgun Mass Spectrometry-Based Proteomic Experiments with Tandem Mass Tag (TMT) Labeling*. 2019, R package version 1.2.7.

[112] G. D. Ruxton, "The unequal variance t-test is an underused alternative to Student's t-test and the Mann–Whitney U test," *Behavioral Ecology*, vol. 17, no. 4, pp. 688–690, Jul. 1, 2006.

[113] J. D. Storey and R. Tibshirani, "Statistical significance for genomewide studies," *PNAS*, vol. 100, no. 16, pp. 9440–9445, Aug. 5, 2003.

[114] T. Saito and M. Rehmsmeier, "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets," *PLoS One*, vol. 10, no. 3, Mar. 4, 2015.

[115] Z. He, T. Huang, X. Liu, P. Zhu, B. Teng, and S. Deng, "Protein inference: A protein quantification perspective," *Computational Biology and Chemistry*, APBC2016, vol. 63, pp. 21–29, Aug. 1, 2016.

[116] L. Jacob, F. Combes, and T. Burger, "PEPA test: Fast and powerful differential analysis from relative quantitative proteomics data using shared peptides," *bioRxiv*, p. 158 212, Jun. 30, 2017.

[117] *Stan.jl: The Julia interface to Stan.*

[118] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A Fresh Approach to Numerical Computing," *CoRR*, vol. abs/1411.1607, 2014.

[119] L. Käll, J. D. Storey, M. J. MacCoss, and W. S. Noble, "Posterior Error Probabilities and False Discovery Rates: Two Sides of the Same Coin," *Journal of Proteome Research*, vol. 7, no. 1, pp. 40–44, Jan. 2008.

[120] M. E. Ritchie, B. Phipson, D. Wu, Y. Hu, C. W. Law, W. Shi, and G. K. Smyth, "Limma powers differential expression analyses for RNA-sequencing and microarray studies," *Nucleic Acids Res*, vol. 43, no. 7, e47–e47, Apr. 20, 2015.

[121] M. Beck, A. Schmidt, J. Malmstroem, M. Claassen, A. Ori, A. Szymborska, F. Herzog, O. Rinner, J. Ellenberg, and R. Aebersold, "The quantitative proteome of a human cell line," *Mol Syst Biol*, vol. 7, p. 549, Nov. 8, 2011.

[122] J. K. Kruschke, "Bayesian estimation supersedes the t test," *Journal of Experimental Psychology: General*, vol. 142, no. 2, pp. 573–603, 2013.

[123] M. J. Betancourt and M. Girolami, "Hamiltonian Monte Carlo for Hierarchical Models," Dec. 3, 2013.

[124] J. M. Pratt, D. M. Simpson, M. K. Doherty, J. Rivers, S. J. Gaskell, and R. J. Beynon, "Multiplexed absolute quantification for proteomics using concatenated signature peptides encoded by QconCAT genes," *Nature Protocols*, vol. 1, no. 2, pp. 1029–1043, Aug. 2006.

[125] M. M. Shariati, I. R. Korsgaard, and D. Sorensen, "Identifiability of parameters and behaviour of MCMC chains: A case study using the reaction norm model," *Journal of Animal Breeding and Genetics*, vol. 126, no. 2, pp. 92–102, 2009, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1439-0388.2008.00773.x.

[126] R. M. Neal, "Slice Sampling," *The Annals of Statistics*, vol. 31, no. 3, pp. 705–741, 2003.

[127] D. B. Rubin, "Estimation in Parallel Randomized Experiments," *Journal of Educational Statistics*, vol. 6, no. 4, pp. 377–401, 1981.

[128] A. O'Hagan and J. Forster, *Kendall's Advanced Theory of Statistics, Vol. 2B: Bayesian Inference*, 2nd ed. London : New York: Edward Arnold ; Halsted Press, 2004.