proDA: Probabilistic Dropout Analysis for Identifying Differentially Abundant Proteins in Label-Free Mass Spectrometry

Constantin Ahlmann-Eltze and Simon Anders*
Center for Molecular Biology (ZMBH), University of Heidelberg, Germany

(Dated: 4 June 2019)

Abstract

Protein mass spectrometry with label-free quantification (LFQ) is widely used for quantitative proteomics studies. Nevertheless, well-principled statistical inference procedures are still lacking, and most practitioners adopt methods from transcriptomics. These, however, cannot properly treat the principal complication of label-free proteomics, namely many non-randomly missing values.

We present *proDA*, a method to perform statistical tests for differential abundance of proteins. It models missing values in an intensity-dependent probabilistic manner. proDA is based on linear models and thus suitable for complex experimental designs, and boosts statistical power for small sample sizes by using variance moderation. We show that the currently widely used methods based on ad hoc imputation schemes can report excessive false positives, and that proDA not only overcomes this serious issue but also offers high sensitivity. Thus, proDA fills a crucial gap in the toolbox of quantitative proteomics.

Availability: The proDA method is implemented as an open-source R package, available on https://github.com/const-ae/proDA.[1]

1. INTRODUCTION

Label-free quantification (LFQ) is a standard approach used in proteomics mass spectrometry (MS). Due to the similarity of this data type to expression microarray data, analysis methods from that field are commonly used for LFQ-MS. A major difference, however, is the presence of missing values in MS, but not in microarray data.

It is well established that missing values do not occur entirely at random, but more often at low intensities (Choi *et al.* 2015, Lazar *et al.* 2016, Ooijen *et al.* 2017, Välikangas *et al.* 2017). The fraction of missing values varies by experimental design, but it is not uncommon to have more than 50% missing values, especially in affinity purification experiments. This issue hence cannot simply be ignored but needs proper handling, and doing so is a central challenge in statistical analysis of LFQ data, e.g., for identifying proteins which are differentially abundant between conditions. In the last years several method have been proposed to tackle this challenge, most of which rely on imputation, i.e., they simply replace missing values with some number that is deemed realistic.

However, a fundamental problem with imputation is that it obscures the amount of available information: imputed values will be considered as equally certain as actually measured values by any downstream processing (identifying differentially abundant proteins, clustering, quality control). This can invalidate inferential conclusions due to underestimating statistical uncertainty or cause loss of statistical power. Therefore, we propose a probabilistic dropout model that explicitly describes the available information about the missing values.

Figure 1A demonstrates that missingness carries information: observations in proteins with many missing values (red) have a lower intensity than observations in proteins with only one or no missing values (purple). In addition, Figure 1B illustrates that the ratio of these densities forms a curve with sigmoidal shape, clearly showing how the probability of a value being missing depends strongly on overall intensity.

If sample size is limited, substantial gains in statistical power can be gained from using shrinkage estimation procedure for variance estimation ("variance moderation") (Lönnstedt and Speed 2002). This approach is widely used in transcriptomics data analysis, e.g., by the limma package (Smyth 2004). The advantage of using limma or similar approaches for LFQ-MS has been advocated only rather recently (e.g., Kammers *et al.* (2015)). For example, the DEP package (Zhang *et al.* 2018) performs imputation followed by a limma analysis to infer differentially abundant proteins. As stated above, the use of imputation may compromise the validity of limma's statistical inference, and hence, the purpose of the present work is to adapt limma-style inference to account for values missing not at random and so improve power and reliability of differential abundance analysis for LFQ-MS.

A typical analysis of a label-free tandem mass spectrometry experiment consists of a number of steps. First, peaks in the MS1 need to be identified using the corresponding MS2 spectra. Second, the MS1 peaks need to be quantified. In the literature, two approaches are popular for this tasks: spectral counting and peak area integration (Wong and Cagney 2010). Abundant peptides are more often recorded by the MS2, thus the number of MS2 spectra associated with a peptide can be used as a proxy for its abundance (Liu *et al.* 2004). Alternatively, more abundant proteins cause larger peaks in the MS1, thus a second approach is to integrate the peak area of a peptide (Bon-

*Electronic address: sanders@fs.tum.de

[1] Submission to Bioconductor is pending.

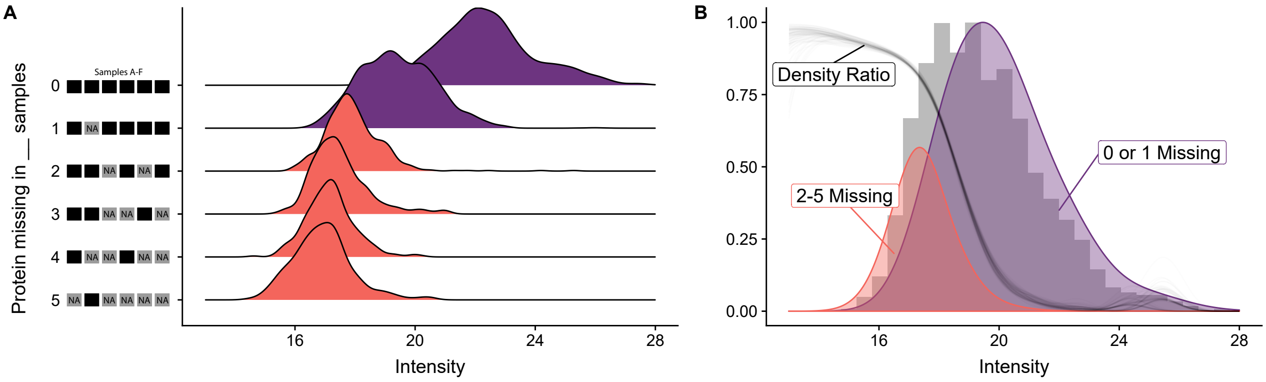


FIG. 1: Missingness in label-free mass spectrometry is informative. (A) Intensity distribution for six replicates of the de Graaf dataset discussed in section 3. Shown is a ridgeline density plot of all the observed intensity values. They have been stratified by the number of samples in which the protein's value was missing. The height of the individual densities is normalized per stratum. Panel (B) shows, in gray, a histogram of all the intensities. Overlayed are densities combining either the values from proteins with at most one missing value (purple) or with more then one missing value (red). The ratio of these two densities (gray line) is has sigmoidal shape. The density ratio has been bootstrapped 100 times to show its sampling distribution.

darenko *et al.* 2002, Chelius and Bondarenko 2002). Subsequent comparisons of the methods by Grossmann *et al.* (2010) and Dowle *et al.* (2016) concluded that peak area based methods perform better than spectral counting. Consequently, we will only focus on methods that handle continuous intensities. The third important step is the aggregation of the peptide level information to protein information. One popular method, that is directly integrated in the popular MaxQuant platform (Cox and Mann 2008), is called MaxLFQ (Cox *et al.* 2014). It combines the peptide intensities across samples using their ratios and has been shown to be highly accurate (Al Shweiki *et al.* 2017, Valikangas *et al.* 2017). The result of all those steps is a table with intensities for each protein and sample. The values in this table are commonly $\log_2$ transformed to account for the mean-variance relationship of the raw data (Supplementary Figure S1).

There are already some methods available to further analyze this table and identify differentially abundant proteins. Perseus (Tyanova *et al.* 2016) is a platform with graphical use interface, developed by the same group as MaxQuant, which provides functionality to normalize the data, impute missing values, identify significant proteins using a t-test and visualize the results. For multiple testing correction, Perseus offers two options: either the Benjamini-Hochberg procedure (Benjamini and Hochberg 1995) or significance analysis of microarrays (SAM), a permutation-based correction originally developed by Tusher *et al.* (2001). As already mentioned, DEP (Zhang *et al.* 2018) is an R package that provides a similar set of functionalities, but uses the more powerful variance moderated t-test to identify significant proteins using the R package limma (Smyth 2004, Smyth *et al.* 2010). For multiple testing correction DEP uses by default the methods in the fdrtool package (Strimmer 2008). In addition, it provides a simple interface to a large number of imputation methods from the MSnbase R package (Gatto and Lilley 2012). In contrast, Perseus only provides two imputation methods, which either replace the missing values with a small deterministic value (MinDet) or with random values jittered around that small value (MinProb). QPROT (Choi *et al.* 2015) is a command line tool that fits, like limma, an empirical Bayesian model. It avoids imputation and instead integrates out the position of missing values using a cumulative normal distribution below a hard limit of detection.

Here, we present proDA (inference of *pro*tein *d*ifferential *a*bundance by *pro*babilistic *d*ropout *a*nalysis). In the following section, we will explain the intuition behind proDA and how it differs from the existing tools. In the third section, we will use a semi-synthetic dataset to compare the performance of the tools and check if they control the false discovery rate (FDR). We will show that proDA shows strong advantages over the existing approaches. In the fourth section, we will use proDA to analyze a real dataset studying ubiquitination, before we close with a discussion of the advantages and limitations of our method.

2. APPROACH

The core of our idea is to combine the sigmoidal dropout curve for missing values with the information from the observed values. Figure 2 gives a conceptual overview of our approach. All the mathematical details of our method are described in Appendix A, where we develop the approach for full linear models. Here, in the main text, we aim to provide a more intuitive explanation. We will first discuss the simple setting of an experiment with only a single condition with 3 replicates, and afterwards discuss inference of differential abundance between two conditions.

We assume that, within one condition, there is one

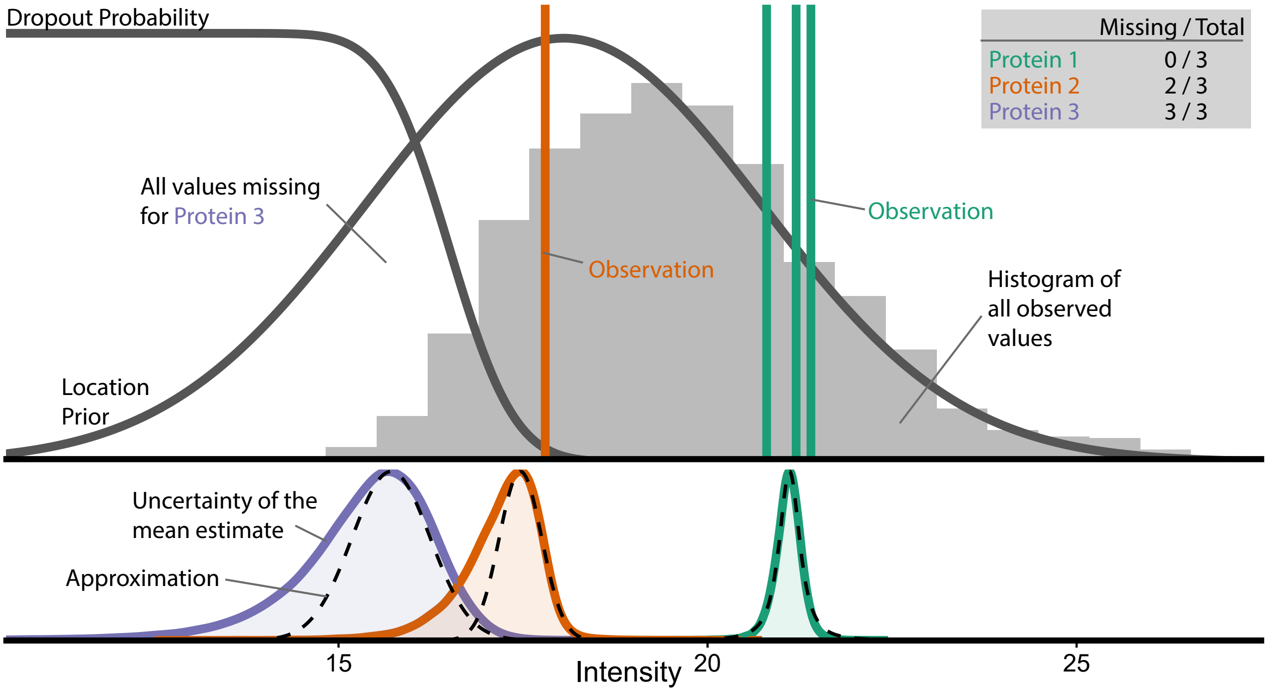


FIG. 2: Intuition behind the probabilistic dropout model. We assume that the real intensity values approximately follow a normal distribution ("location prior"). The probability of not observing one of these values ("dropout probability") is high for low intensity and low for high intensity. Hence, the distribution of the actually observed values (gray histogram) is skewed, with values missing in its left flank. The vertical lines indicate the observed intensities for three hypothetical proteins: Protein 1 (green) has complete observations, protein 2 (orange) has 2 missing values, and protein 3 (purple) has only missing values. The lower panel shows the inferred posterior probability distribution for the means for proteins 1, 2, and 3 (calculated using Stan (Stan Development Team 2017)). The dashed lines show the symmetric approximation to these that we use for efficient inference.

expected value for each protein, the population average, i.e., the mean value we would get if we averaged over infinitely many replicate samples. The abundances in our 3 replicates scatter around this unknown "true" mean value, and our goal is to infer a posterior distribution that contains the true mean and captures our uncertainty about its location. In case of no missing values, such a posterior takes the shape of the t distribution, which is the basis for the well known Student's t test (green posterior in Figure 2). Missing values cause these posteriors to become skewed , wider, and their mode (peak) to shift to the left of the average of the observed values (because the missing values are likely lower than the observed ones); see orange posterior in Figure 2. Even with no observed values, we can infer a posterior (purple posterior in the figure): its left flank follows the location prior, i.e., the distribution of values we actually expect in our data, and its right flank follows the dropout curve, because higher values would have likely been observed.

Hence, our approach first estimates from the data for all proteins the shape of the dropout probability curve and of the location prior. It then uses this information to infer for each protein an approximate posterior for its mean, with the necessary shift in mode location and widening due to the additional uncertainty from any missing values. We approximate the skewed posteriors with a symmetric approximation (dashed lines in the figure) that follows the right flank. This improves performance and is permissible because the flank on the lower side is irrelevant for inference of difference.

The process involves so-called shrinkage estimation (or moderation), which shares information across proteins in order to improve variance estimation (as originally proposed by Lönnstedt and Speed (2002) and also used in limma (Smyth 2004)). Furthermore, we apply shrinkage estimation not only to the variance but also to the location, as this enables us to handle the edge case of all observation missing in one condition.

To test a protein for differential abundance between two conditions, we compare the approximate posteriors inferred for the two conditions and calculate a p value for the null hypothesis of both true means being equal. We do this using linear models, which allows for accommodating known covariates and complex experimental designs, in the same manner as limma offers for transcriptomics experiments.

3. VALIDATION AND COMPARISON

We validated our approach and compared it with the existing methods discussed in the introduction. In order to evaluate performance with data that is

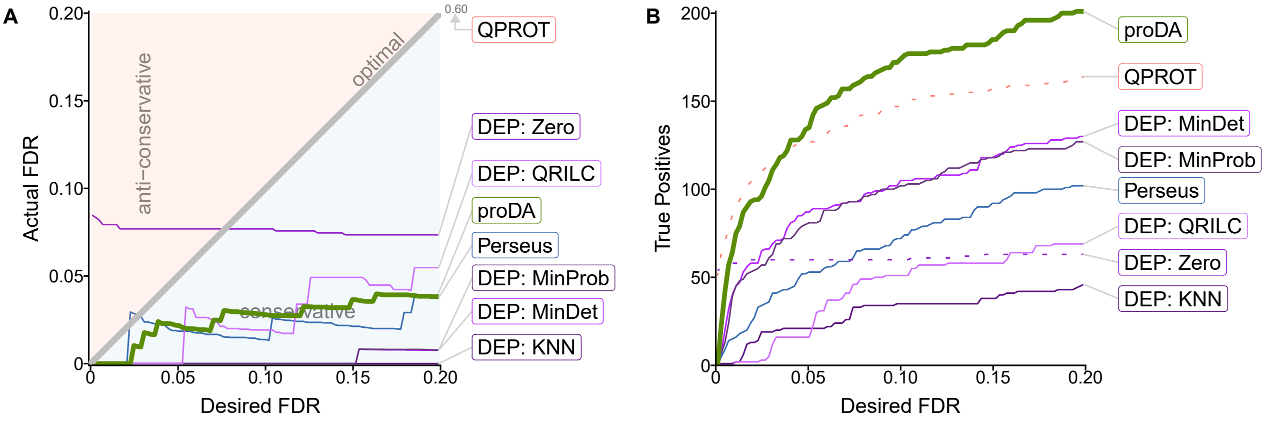


FIG. 3: Performance comparison on the de Graaf dataset with three against three samples and 20% changed proteins. A) Comparison of the user specified FDR (cut-off on BH-adjusted p value) with the FDR that is actually achieved by the tool according to the ground truth. The line for the QPROT method is missing because it is literally off the chart, with an average FDR of 60%. B) Plot showing how many actually changed proteins (true positives) each method identified at a specified FDR level. The two methods shown as dotted lines should be considered “disqualified” as they failed to control FDR in panel A.)

realistic, yet still has known ground truth, we will use a data set comprising only replicates of a single condition, so that we can expect there to be no differentially abundant proteins. We will then introduce differences artificially for some proteins so that we know the ground truth. de Graaf *et al.* (2014) analysed phosphorylation dynamics in Jurkat T cells over 6 time points using affinity purification. We only use the first time point, for which there are 18 samples, 3 biological replicates with 3 technical replicates each, which were measured in two separate mass spectrometry runs. Supplementary Figure S2 shows a heatmap of the data. There are many missing values (49%), which helps us to asses their impact on the different methods.

In a typical affinity purification experiment, it is not unusual to have only three replicates per condition. So, we chose six samples and divided them into two synthetic conditions, ensuring that both contain a mix of different biological replicates, so that there is no signal in the null dataset (row marked “3v3” in Supplementary Figure S2). In the next step, we select 20% of all proteins and randomly shuffle those rows, but only in the first condition. This creates a realistic dataset where we know which proteins differ between condition one and two. Other approaches, where a selected number of proteins are shifted by a fixed effect size, are not applicable, because shifting the mean of a protein would also imply a different probability for missing observations.

We compare the four methods discussed in the introduction (proDA, DEP, QPROT, and Perseus), running each tool with their default settings, except for the multiple testing correction, where we always use Benjamini-Hochberg’s method to make the results more comparable. The R markdown notebook used to conduct the tests is available on github.com/const-ae/proDA-Paper. DEP offers a range of different imputation methods; we chose to test it with five typical ones: Zero, MinDet, MinProb, KNN, and QRLIC. We ran QPROT with 2000 burn-in and 10,000 sampling iterations.

Figure 3 shows the performance of the tools: For each cut-off value on the Benjamini-Hochberg adjusted p values (desired false discovery rate (FDR)), we calculate the actual false discovery rate using the ground truth. In the optimal case, both would be identical and the lines of the tools would be on the diagonal. If a methods gets above the diagonal line, this means that it called too many false positives (failed to control type-I error), which is highly problematic. In our test, most methods, including proDA succeed in controlling FDR, with the exception of QPROT and DEP with imputation mode Zero.

For those methods that passed the FDR control requirement, we can now ask which has most inferential power. Figure 3B shows the number of true positives that each method recovered depending on the desired FDR. proDA performs well in this test. Its actual FDR always stays below the desired FDR and at 10% desired FDR, it recovers 65% more true positives than the second best approach, DEP with MinDet imputation. The performance of DEP depended on the imputation method that is used. Zero imputation is problematic, as can be seen in this example, because it fails to control the FDR at small values. The best imputation methods are MinDet and MinProb, which perform nearly identical. Perseus with the MinProb imputation recovers fewer true positives than DEP, which is expected, because it uses the classical t-test and not the variance moderated version. QPROT consistently fails to control the FDR. At an desired FDR of 10%, it calls a total of 363 proteins significant, of which 216 are false positives (i.e., the actual FDR is 60%).

In Supplementary Figure S3, we further distinguish the calibration and performance by the number of actually observed values in condition one and two. This shows that QPROT is unable to control the FDR, because it has many false positives for proteins with zero against one observations. proDA is more powerful than the other methods, because it shows consistently good performance across comparisons and in particular if only one or two observations are missing. The zero imputation methods always identifies all zero against three observations as significant. In many cases this is correct, but as this does not depend on the desired FDR, this can lead to an actual FDR that is too large if the user specifies a small FDR.

In Supplementary Figure S4-S6, we show that we get consistent results even if we change the number of compared samples (3 vs 3, 4 vs 4, and 6 vs 6) and also if we change the percentage of true positive proteins (5%, 10%, 20%, and 30%), although the degree by which proDA outperforms the other tools seems to dependent on the number of missing values.

4. APPLICATION

After demonstrating that proDA controls the FDR and is able to recover the largest number of changed proteins, we applied it to analyze a dataset on the interaction landscape of ubiquitin (Zhang *et al.* 2017). In this example, we do not know the ground truth, but show that we recover proteins that biologically make sense. In the original publication the authors analyzed the dataset using Perseus, later they presented the DEP R package for analyzing such datasets (Zhang *et al.* 2018). We will re-run the analysis that Zhang et al. describe with proDA.

Ubiquitin is a small protein that plays an important role in many different signaling pathways. There are three different kinds of ubiquitination: mono-ubiquitination, multi-mono-ubiquitination and poly-ubiquitination. Poly-ubiquitination is further distinguished by the linkage between the donor and the acceptor ubiquitin. The donor is linked with its C terminus to any of the seven lysines (K6, K11, K27, K29, K33, K48, K63) or the terminal methionine (M1) of the acceptor. Zhang et al. studied the recognition of those eight linkages and mono-ubiquitin by ubiquitin binding proteins. For this, they developed a new technique called ubiquitin interactor affinity enrichment-mass spectrometry (UbIA-MS) (Zhang *et al.* 2018).

They run an enrichment experiment for each of the eight ubiquitin linkages plus one condition with mono-ubiquitin (Mono) and one empty control condition (ctrl). Each condition was measured in triplicates. To determine which proteins bind (directly or indirectly) to any of the ubiquitin linkages, we always compare the intensity for each protein to the corresponding intensity in the control group.

Figure 4 shows the results of the analysis with proDA. Figure 4A compares the total number of significant interactors at a nominal FDR of 10%, filtering out all proteins that had higher intensity in the control condition than in the ubiquitin condition. Figure 4B further stratifies the data from panel A. It not just describes how many proteins bind to a linkage, but also how many proteins bind to a specific combination of linkages. We can see that a majority interacts significantly with with all ubiquitins, but there are also proteins showing significant interactions only for specific linkages.

Figure 4D demonstrates that proDA has not just recovered many interactors, but proteins related to gene ontology sets relevant for ubiquitination (Ashburner *et al.* 2000, Carbon *et al.* 2019, Yu *et al.* 2012). In addition to the 9 ubiquitination conditions, here we also list the results of conducting an F test to identify all proteins that differ in any condition, as an example for the ability of proDA to perform missing value aware ANOVA.

5. DISTANCES

A commonly used approach for sample quality control is to calculate some measure of similarity for all pairs of samples, in order to check whether replicate samples appear more similar than samples from different conditions.

Typically, Euclidean distance is used, e.g. by Zhang *et al.* (2018) who use MinProb imputation before Euclidean distance is calculated. Figure 4C shows the outcome of this procedure for the ubiquitin data. Differences in the shape of the dropout probability curve can strongly influence a distance calculated in this manner. Based on the proDA model, we developed an approach to calculate Euclidean distance in a probabilistic manner without the need for imputation in order to reduce the effect of differences in dropout probabilities (Appendix B). In fact, our distance calculation is able to recover the triplet structure of the data set, while the MinProb imputation based distances do not (Figure 4C).

6. CONCLUSION

In this paper, we have presented our R package proDA for identifying proteins that are differentially abundant in label-free mass spectrometry data sets. The main challenge for analyzing label-free mass spectrometry data are the large number of missing values. We suggest to handle them using a probabilistic dropout model combined with empirical Bayesian priors to combine the available information from observed and missing values.

In the performance comparison with existing tools on a semi-synthetic data set with known ground truth, we saw that proDA recovers more true positives, while controlling the false discovery rate. We showed that imputation can be problematic because it either leads to a loss of power or worse to not controlling the false discovery rate. The improved sensitivity of proDA comes at the prize of a somewhat increased run time.

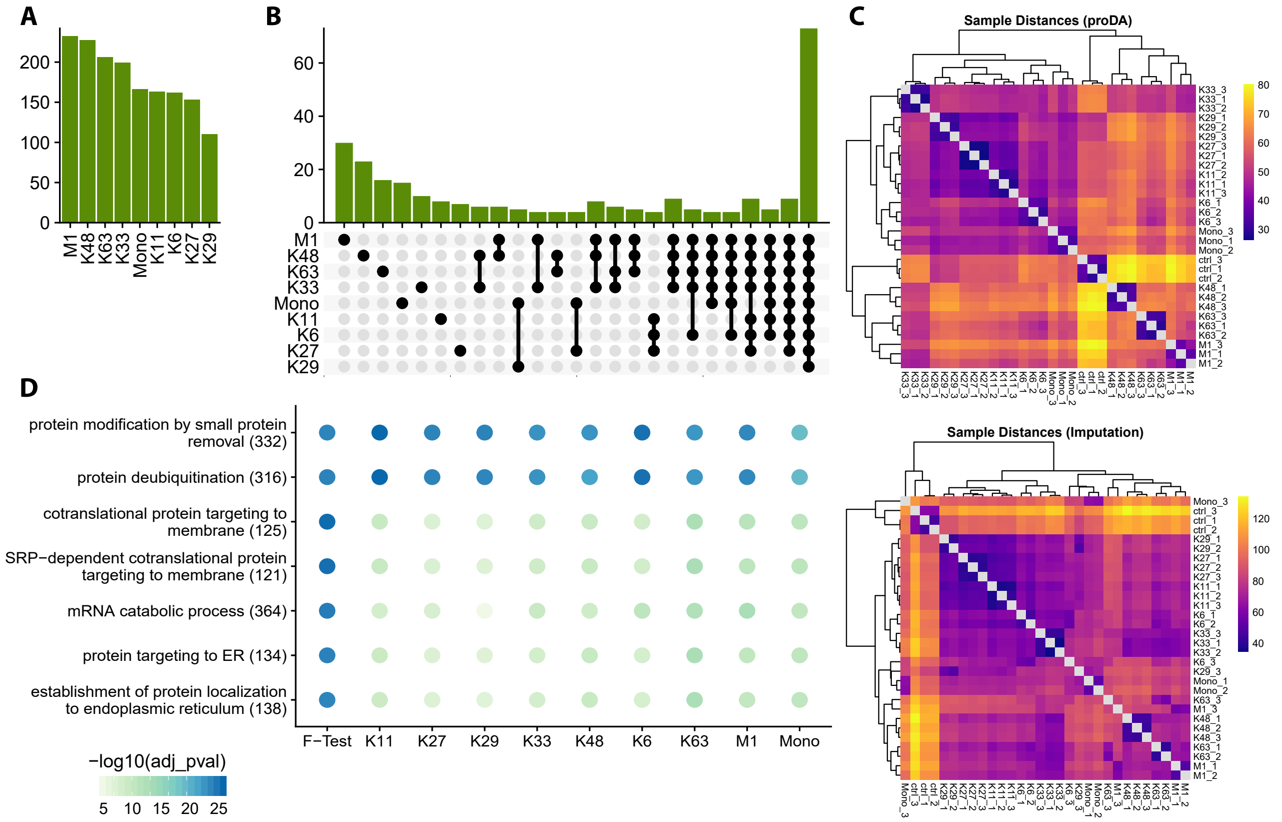


FIG. 4: Ubiquitination analysis with proDA. A) shows the number of interactors for each of the nine conditions. The green bars show the number of proteins identified as significant interactors for each condition with proDA. B) breaks down those interactors into more detail. It shows the number of proteins that interact with a specific combination of ubiquitin linkages. The total number of intersections was limited to the largest 25 sets ordered by degree. C) shows two heatmaps with the sample distances, calculated according to Equation (B4) (upper heatmap) and on the imputed dataset (lower heatmap). The rows and columns were clustered using hierarchical clustering on the distances. D) shows a dot plot with the seven most significant gene ontology (GO) terms related to the set of interactors with any of the nine conditions and the set of proteins that differ over all conditions ("F-test").

Whereas the imputation based methods finish within seconds, our model might need one or two minutes to calculate a result. In the end, we believe the increased computational demands are justified, because the analysis run time is still fast enough for interactive use.

In conclusion, we have demonstrated that imputation can be problematic and that properly modelling the uncertainty posed by missing values boosts power.

Funding

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) via SFB 1036.

Al Shweiki, M. H. *et al.* (2017). Assessment of Label-Free Quantification in Discovery Proteomics and Impact of Technological Factors and Natural Variability of Protein Abundance. *Journal of Proteome Research*, **16**(4), 1410–1424.

Ashburner, M. *et al.* (2000). Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, **25**(1), 25–9.

Benjamini and Hochberg (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Statist Soc B*, **57**(1), 289–300.

Bondarenko, P. V. *et al.* (2002). Identification and relative quantitation of protein mixtures by enzymatic digestion followed by capillary reversed-phase liquid chromatography - Tandem mass spectrometry. *Analytical Chemistry*, **74**(18), 4741–4749.

Broyden, C. G. (1970). The Convergence of a Class of Double-rank Minimization Algorithms. *IMA Journal of Applied Mathematics*, **6**(1), 76–90.

Carbon, S. *et al.* (2019). The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Research*, **47**(D1), D330–D338.

Chelius, D. and Bondarenko, P. V. (2002). Quantitative profiling of proteins in complex mixtures using liquid chromatography and mass spectrometry. *Journal of Proteome Research*, **1**(4), 317–323.

Choi, H. *et al.* (2015). QPROT: Statistical method for testing differential expression using protein-level intensity data in label-free quantitative proteomics. *Journal of Proteomics*, **129**, 121–126.

Cox, J. and Mann, M. (2008). MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology*, **26**(12), 1367–1372.

Cox, J. *et al.* (2014). Accurate Proteome-wide Label-free Quantification by Delayed Normalization and Maximal Peptide Ratio Extraction, Termed MaxLFQ. *Molecular & Cellular Proteomics*, **13**(9), 2513–2526.

de Graaf, E. L. *et al.* (2014). Single-step Enrichment by Ti4 + -IMAC and Label-free Quantitation Enables In-depth Monitoring of Phosphorylation Dynamics with High Reproducibility and Temporal Resolution . *Molecular & Cellular Proteomics*, **13**(9), 2426–2434.

Dowle, A. A. *et al.* (2016). Comparing the Diagnostic Classification Accuracy of iTRAQ, Peak-Area, Spectral-Counting, and emPAI Methods for Relative Quantification in Expression Proteomics. *Journal of Proteome Research*, **15**(10), 3550–3562.

Dunn, P. K. and Smyth, G. K. (2018). *Generalized Linear Models With Examples in R*.

Efron, B. and Morris, C. (1975). Data Analysis Using Stein's Estimator and its Generalizations. *1Journal of the American Statistical Association*, **70**(350), 311–319.

Ellison, B. E. (1964). Two Theorems for Inferences about the Normal Distribution with Applications in. Technical Report 305.

Fletcher, R. (1970). A new approach to variable metric algorithms. *The Computer Journal*, **13**(3), 317–322.

Gatto, L. and Lilley, K. (2012). MSnbase - an R/Bioconductor package for isobaric tagged mass spectrometry data visualization, processing and quantitation. *Bioinformatics*, **28**, 288–289.

Gay, D. M. (1990). Usage summary for selected optimization routines. *AT&T Bell Laboratories, Murray Hill, NJ 07974*, (153), 1–21.

Goldfarb, D. (1970). A family of variable-metric methods derived by variational means. *Mathematics of Computation*, **24**(109), 23–23.

Grossmann, J. *et al.* (2010). Implementation and evaluation of relative and absolute quantification in shotgun proteomics with label-free methods. *Journal of Proteomics*, **73**(9), 1740–1746.

Kammers, K. *et al.* (2015). Detecting significant changes in protein abundance. *EuPA Open Proteomics*, **7**, 11–19.

Lazar, C. *et al.* (2016). Accounting for the Multiple Natures of Missing Values in Label-Free Quantitative Proteomics Data Sets to Compare Imputation Strategies. *Journal of Proteome Research*, **15**(4), 1116–1125.

Liu, H. *et al.* (2004). A model for random sampling and estimation of relative protein abundance in shotgun proteomics. *Analytical Chemistry*, **76**(14), 4193–4201.

Lönnstedt, I. and Speed, T. (2002). Replicated microarray data. *Statistical Sinica*, **12**(12), 31–46.

Mathai, A. and Provost, S. (1992). Quadratic Forms in Random Variables.

Ooijen, M. P. V. *et al.* (2017). Identification of differentially expressed peptides in high-throughput proteomics data. *Briefings in Bioinformatics*, **1**(February), 1–11.

Shanno, D. F. (1970). Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, **24**(111), 647–647.

Smyth, G. K. (2004). Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments. *Statistical Applications in Genetics and Molecular Biology*, **3**(1), 1–26.

Smyth, G. K. *et al.* (2010). Linear Models for Microarray Data User's Guide (limma).

Stan Development Team (2017). Stan Modeling Language. *User Guide and Reference Manual*, pages 1–188.

Strimmer, K. (2008). fdrtool: A versatile R package for estimating local and tail area-based false discovery rates. *Bioinformatics*, **24**(12), 1461–1462.

Tusher, V. G. *et al.* (2001). Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, **98**(9).

Tyanova, S. *et al.* (2016). The Perseus computational platform for comprehensive analysis of (prote)omics data. *Nature Methods*, **13**(9), 731–740.

Välikangas, T. *et al.* (2017). A comprehensive evaluation of popular proteomics software workflows for label-free proteome quantification and imputation. *Briefings in Bioinformatics*, **1**(April), 1–12.

Valikangas, T. *et al.* (2017). Comprehensive evaluation of popular proteomics software workflows for label-free proteome quantification and imputation. *Briefings in Bioinformatics*, **19**(2), 185–193.

Wong, J. W. H. and Cagney, G. (2010). An overview of label-free quantitation methods in proteomics by mass spectrometry. In *Proteome bioinformatics*, pages 273–283. Springer.

Yu, G. *et al.* (2012). clusterProfiler: an R package for comparing biological themes among gene clusters. *Omics: a journal of integrative biology*, **16**(5), 284–287.

Zacks, S. (1981). *Parametric statistical inference: basic theory and modern approaches*, volume 4. Elsevier.

Zhang, X. *et al.* (2017). An Interaction Landscape of Ubiquitin Signaling. *Molecular Cell*, **65**(5), 941–955.e8.

Zhang, X. *et al.* (2018). Proteome-wide identification of ubiquitin interactions using UbIA-MS. *Nature Protocols*, **13**(3), 530–550.

Appendices

Appendix A: Mathematical Description of the Probabilistic Dropout Model

At the center of proDA is the idea of a probabilistic dropout model. As we saw in Figure 1, the chance of a missing value decreases with increasing protein intensity. We will model this relationship with a sample specific sigmoid dropout curve. To better understand our model, we will describe it as a generative model, ie. the mathematical relations that we believe could be responsible for the data table from which we start our analysis.

We will call this table a matrix Y with $I \times J$ rows and columns, where I is the number of proteins and J is the number of samples. The information about each sample is encoded in a model matrix X with J rows and p columns. Our goal is to find for each protein the p coefficients of the vector $\boldsymbol{\beta}_i$. For each sample and protein we define the predicted value as $\hat{\boldsymbol{\mu}}_i = X\hat{\boldsymbol{\beta}}_i$.

We assume that for each protein $i \in \{1, \dots, I\}$ the observations y_{ij} are drawn from a Normal distribution with mean μ_{ij} and variance σ_i^2. However, not every y_{ij} is observed; importantly, some are missing ($y_{ij} = \texttt{NA}$). As we have previously discussed, we assume that the probability of a dropout depends on the underlying intensity and we will model this with a sigmoidal relationship. There are several possible functions describing curves with sigmoidal shape; for mathematical convenience, we chose the inverse probit, i.e., the cumulative density function (CDF) of a Normal distribution. In formal notation, this model is

$$\begin{aligned} \mu_{ij} &= X_j \boldsymbol{\beta}_i \\ z_{ij}|\mu_{ij}, \sigma_i^2 &\sim \text{Normal}(\mu_{ij}, \sigma_i^2) \\ d_{ij}|z_{ij}, \rho_j, \zeta_j &\sim \text{Bernoulli}(\Phi(z_{ij}; \rho_j, \zeta_j^2)) \\ y_{ij}|z_{ij}, d_{ij} &= \begin{cases} \texttt{NA}, & \text{if } d_{ij} \\ z_{ij}, & \text{else.} \end{cases} \end{aligned} \tag{A1}$$

Here, z_{ij} are the latent intensities, that we do not have full access to because of the dropouts. $\boldsymbol{\beta}_i$ are the coefficients for which we want to find out if they or their linear combination are different from zero. d_{ij} indicates if a protein is missing in the specific sample. The probability of missingness ($z_{ij} = 1$) is given by the sigmoidal dropout curve $\Phi(\cdot)$ (Normal CDF), which is parameterized using the inflection point ρ_j and the scale ζ_j:

$$\Phi(x; \rho, \zeta^2) = \frac{1}{\sqrt{2\pi}\zeta} \int_{-\infty}^{x} \exp\left(-\frac{(t-\rho)^2}{2\zeta^2}\right) dt. \tag{A2}$$

In addition, we assume that the means μ_{ij} and the variances σ_i^2 are similar across proteins and add the priors

$$\mu_{ij}|\mu_0, \sigma_0^2 \sim \text{Student-t}(\text{df}_{\text{loc}}, \mu_0, \sigma_0^2) \tag{A3}$$

and

$$\sigma_i^2|\text{df}_0, \tau_0^2 \sim \text{Scaled-inv-}\chi^2(\text{df}_0, \tau_0^2). \tag{A4}$$

The prior in Equation (A3) on the protein means μ_{ij} is important to handle the edge case if in one condition a protein is completely missing. The prior in Equation (A4) corresponds to the variance moderation of limma (Smyth 2004).

The probability density function of the generalized Student's t-distribution is defined as

$$f_t(x; \text{df}_{\text{loc}}, \mu, \sigma^2) = \frac{\Gamma\left(\frac{\text{df}_{\text{loc}}+1}{2}\right)}{\Gamma\left(\frac{\text{df}_{\text{loc}}}{2}\right)\sqrt{\pi \text{df}_{\text{loc}} \sigma^2}} \left(1 + \frac{1}{\text{df}_{\text{loc}}} \frac{(x-\mu)^2}{\sigma^2}\right)^{-\frac{\text{df}_{\text{loc}}+1}{2}} \tag{A5}$$

and the probability density function of the scaled inverse χ^2 distribution is

$$f_{\text{Inv-}\chi^2}(x; \tau^2, \text{df}) = \frac{(\tau^2 \text{df}/2)^{\text{df}/2}}{\Gamma(\text{df}/2)} \frac{\exp\left(-\frac{\text{df}\tau^2}{2x}\right)}{x^{1+\text{df}/2}}. \tag{A6}$$

We iteratively estimate the hyper parameters and the protein specific parameters using a maximum *a posteriori* approach until the model converges. To identify which coefficients in $\boldsymbol{\beta}_i$ are significant, we use a Wald test or likelihood ratio F-test (Dunn and Smyth 2018).

1. Model Fitting

In the following section, we will explain how to infer the feature parameters $\boldsymbol{\beta}_i$ and σ_i^2, and then the hyper-parameters μ_0, σ_0^2, τ_0^2, df_0, $\boldsymbol{\rho}$ and $\boldsymbol{\zeta}$. We assume that $\mathrm{df}_{\mathrm{loc}}$ is fixed by the user.

To simplify the notation, we will first focus on only one protein and assume that all samples belong to the same condition and thus suppress all subscripts i and j. This also allows us to directly talk about μ instead of $X\boldsymbol{\beta}$, because in that specific case they are identical.

If there were no missing values and if we ignored the priors, the likelihood of the mean μ and σ^2 given the observations $\boldsymbol{y}$ would be

$$L(\mu, \sigma^2|\boldsymbol{y}) \propto \prod_j f_{\mathrm{Normal}}(y_j; \mu, \sigma^2). \tag{A7}$$

To handle the mix of observed and missing values in $\boldsymbol{y}$, we will extend the above equation by marginalizing out the missing values

$$\begin{aligned} L(\mu, \sigma^2|\boldsymbol{y}) \propto & \prod_{j: y_j \neq \mathtt{NA}} f_{\mathrm{Normal}}(y_j; \mu, \sigma^2) \\ & \times \prod_{j: y_j = \mathtt{NA}} \int_{-\infty}^{\infty} f_{\mathrm{Normal}}(z; \mu, \sigma^2) \Phi(z; \rho_j, \zeta_j^2) dz. \end{aligned} \tag{A8}$$

The integral in Equation (A8) can be simplified

$$\int_{-\infty}^{\infty} f_{\mathrm{Normal}}(z; \mu, \sigma^2) \Phi(z; \rho, \zeta^2) dz = \Phi(\mu; \rho, \zeta^2 + \sigma^2), \tag{A9}$$

with the proof for example provided by Ellison (1964) and Zacks (1981).

Now, we can combine Equation (A8) and Equation (A9), add the priors that we proposed in Equation (A3) and Equation (A4), and use $X\boldsymbol{\beta}$ instead of μ. We find that the joint density is

$$\begin{aligned} p(\boldsymbol{\beta}, \sigma^2, \mu_0, \sigma_0^2, \mathrm{df}_0, \tau_0^2, \boldsymbol{\rho}, \boldsymbol{\zeta}|\boldsymbol{y}, X) \propto & f_{\mathrm{Inv}\text{-}\chi^2}(\sigma^2; \tau_0^2, \mathrm{df}_0) \\ & \times \prod_j f_{\mathrm{t}}(X_j\boldsymbol{\beta}; \mathrm{df}_{\mathrm{loc}}, \mu_0, \sigma_0^2) \\ & \times \prod_{j: y_j \neq \mathtt{NA}} f_{\mathrm{Normal}}(X_j\boldsymbol{\beta}; x_j, \sigma^2) \\ & \times \prod_{j: y_j = \mathtt{NA}} \Phi(X_j\boldsymbol{\beta}; \rho_j, \sigma^2 + \zeta^2). \end{aligned} \tag{A10}$$

Equation (A10) is the starting point from which we will derive the feature and hyper-parameters. We use a two-step procedure where we first fix the hyper-parameters to estimate the feature parameters and then fix the feature parameters to estimate the hyper-parameters. We iterate between those two steps until the estimates have converged.

2. Feature Parameter Estimation

Given the set of hyper-parameters, we use a maximum *a posteriori* (MAP) approach to find the $\hat{\boldsymbol{\beta}}_i$ and $\hat{\sigma}_i^2$ that best explain the values $\boldsymbol{y}_i$. We take the logarithm of Equation (A10) and derive its Jacobian and Hessian to efficiently find the mode. We use the `nlminb` function in R that wraps the PORT routines (Gay 1990) for the actual optimization.

a. *Unbiased Variance Estimates*

The MAP estimates for the coefficients ($\hat{\boldsymbol{\beta}}$) are already good, but the bias in $\hat{\sigma}^2$ is problematic. In a standard linear model, we would expect the maximum likelihood estimator $\hat{\sigma}^2$ to be biased and underestimate the true variance σ^2 by

$$\mathbb{E}[\hat{\sigma}^2] = \sigma^2 \frac{n-p}{n}, \tag{A11}$$

and hence correct it by multiplying with $n/(n-p)$.

With missing values, the challenge is that we do not know n. The direct approaches of setting $n = J$ or $n = |\{y_j \neq \texttt{NA}\}|$ are problematic because they over- or underestimate the amount of information from the missing values. Instead we will estimate n using the variance of $\hat{\sigma}^2$ at the mode, which is given by

$$\mathbb{V}[\hat{\sigma}^2] = -\frac{1}{\frac{\partial^2 \log p}{\partial(\sigma^2)^2}},$$

where p is the posterior given in Equation (A10). We get this second derivative for free as an element of the Hessian matrix $\boldsymbol{H}$, which is calculated anyway during the maximization of $\log p$:

$$\boldsymbol{H} = \begin{pmatrix} \frac{\partial^2 \log p}{\partial \beta_1^2} & \cdots & \frac{\partial^2 \log p}{\partial \beta_1 \partial \beta_p} & \frac{\partial^2 \log p}{\partial \beta_1 \partial \sigma^2} \\ \vdots & \ddots & \vdots & \vdots \\ \frac{\partial^2 \log p}{\partial \beta_p \partial \beta_1} & \cdots & \frac{\partial^2 \log p}{\partial \beta_p^2} & \frac{\partial^2 \log p}{\partial \beta_p \partial \sigma^2} \\ \frac{\partial^2 \log p}{\partial \sigma^2 \partial \beta_1} & \cdots & \frac{\partial^2 \log p}{\partial \sigma^2 \partial \beta_p} & \frac{\partial^2 \log p}{(\partial \sigma^2)^2} \end{pmatrix} \tag{A12}$$

and that we get for free with the optimization.

We find the value of n using an analogy to the standard linear model without missing values. If we have some values y and use their mean $\bar{y}$ as the mean estimate $\hat{\mu}$, the density of σ^2 would be

$$\begin{aligned} p(\sigma^2|\boldsymbol{y}) &\propto \prod_{i=1}^{n} f_{\text{Normal}}(y_i; \bar{y}, \sigma^2) \\ &\propto \left(\frac{1}{\sigma^2}\right)^{n/2} \exp\left(-\frac{\sum_i (y_i - \bar{y})^2}{2\sigma^2}\right) \\ &\propto f_{\text{Inv-Gamma}}(\sigma^2; \alpha = n/2 - 1, \beta = \text{RSS}/2), \end{aligned} \tag{A13}$$

where $\text{RSS} = \sum_i (y_i - \bar{y})^2$. The mode of the inverse gamma distribution is

$$\text{mode} = \frac{\beta}{\alpha + 1}. \tag{A14}$$

We can now find the expected value of the second derivative that the inverse gamma distribution has at the mode, which is

$$\left.\frac{d^2 f_{\text{Inv-Gamma}}(x, \alpha, \beta)}{(d\sigma^2)^2}\right|_{x=\text{mode}} = \frac{\beta^2}{(\alpha+1)^3}. \tag{A15}$$

With missing values, we can still identify identify the MAP for $\hat{\sigma}^2$ and the associated uncertainty $\mathbb{V}[\hat{\sigma}^2]$. If we now plug in those values

$$\text{mode} = \hat{\sigma}^2 \tag{A16}$$

and

$$\left.\frac{d^2 f_{\text{Inv-Gamma}}(x, \alpha, \beta)}{(d\sigma^2)^2}\right|_{x=\text{mode}} = \mathbb{V}[\hat{\sigma}^2] \tag{A17}$$

and solve Equation (A14) and (A15) using Equation (A13) for n and RSS, we find that

$$\hat{n} = 2\frac{\left(\hat{\sigma}^2\right)^2}{\mathbb{V}[\hat{\sigma}^2]}, \tag{A18}$$

and

$$\widehat{\text{RSS}} = 2\frac{\left(\hat{\sigma}^2\right)^3}{\mathbb{V}[\hat{\sigma}^2]}. \tag{A19}$$

Finally, we can now identify the unbiased estimate of the variance, which is

$$\hat{s}^2 = \frac{\widehat{\text{RSS}}}{\hat{n} - p} \tag{A20}$$

and estimate the degrees of freedom

$$\widehat{\text{df}} = \hat{n} - p. \tag{A21}$$

Sometimes $\hat{n} < p$, in which case we fix $\widehat{\text{df}}$ to a small, but positive value (ie. 0.001), and estimate

$$\hat{s}^2 = \sqrt{\mathbb{V}[\hat{\sigma}^2]\frac{(\widehat{\text{df}} + p)^3}{2\widehat{\text{df}}^2}} \tag{A22}$$

so that the approximation matches the scale of the original distribution, although the mode is slightly off.

b. Variance of the Coefficient Estimates

In a standard linear model, it is easy to find the standard error for the coefficients because it is just

$$\mathbb{V}[\hat{\boldsymbol{\beta}}] = \hat{s}^2 \left(X'X\right)^{-1}. \tag{A23}$$

Again, this cannot be directly applied to the case with missing values, because it is possible that $\hat{s}^2$ is small, but we nevertheless are very uncertain of β_j because there are many missing values for that coefficient. Instead, we will therefore use the inverse of the Hessian of the coefficients, which we calculate using the unbiased estimate $\hat{s}^2$

$$\mathbb{V}[\hat{\boldsymbol{\beta}}]_{\sigma^2=\hat{s}^2} = -(\boldsymbol{H}_{\sigma^2=\hat{s}^2})^{-1}. \tag{A24}$$

This works well for the cases where the distribution of β_j does not have too much skew. But in Figure 2, we saw that for the cases with many missing values the skew can be considerable. If we would just use the matched variance at the mode, we would be wrong on both sides of the distribution. On the left, the approximation would be too narrow and on the right, it would be too wide.

We know that the distributions with considerable skew are always on the low end of the intensity distribution. Thus, in the typical comparison it is most important to get the right flank of the distribution correct, in order to not unnecessarily lose power. We will calculate a correction factor that reduces the variance in order to match the right flank of the distribution. If there is no skew, we know that if we go k units from the mode $\hat{\boldsymbol{\beta}}$ in the direction of β_i the log probability should decrease by $\frac{k}{2\mathbb{V}[\beta_i]}$, because the log density should behave like a multivariate parabola. Note that we still use the Hessian with $\sigma^2 = \hat{\sigma}^2$.

From this relation we can calculate the correction factor which is

$$\text{cf}_{\beta_j} = \frac{k}{2(\log p(\boldsymbol{\beta}; \hat{\sigma}^2, \cdot) - \log p(\boldsymbol{\beta} + \boldsymbol{\beta}_{\text{shift}}; \hat{\sigma}^2, \cdot))} \tag{A25}$$

where $\boldsymbol{\beta}_{\text{shift}}$ is a vector of zeros, except for j's entry which is

$$\beta_{\text{shift}_j} = \sqrt{k(\mathbb{V}[\hat{\boldsymbol{\beta}}]_{jj} - \mathbb{V}[\hat{\boldsymbol{\beta}}]_{j,-j}\mathbb{V}[\hat{\boldsymbol{\beta}}]^{-1}_{-j,-j}\mathbb{V}[\hat{\boldsymbol{\beta}}]_{-j,j})} \tag{A26}$$

We then identify the final covariance matrix as

$$\Sigma = \mathbb{V}[\hat{\boldsymbol{\beta}}]^{(\text{corr})}_{\sigma^2=\hat{s}^2} = \text{diag}(\sqrt{\mathbf{cf}})\mathbb{V}[\hat{\boldsymbol{\beta}}]_{\sigma^2=\hat{s}^2}\text{diag}(\sqrt{\mathbf{cf}}), \tag{A27}$$

where **cf** is the vector formed by the correction factors from Equation (A25).

3. Hyper-parameter Estimation

In the previous section, we have focused on individual proteins and suppressed the subscript i and handled $\boldsymbol{y}$ as vector of size j. Now, we will describe how to fit the hyper-parameters across proteins and thus mention i and work with the full data matrix Y.

a. Dropout Curves

We usually fit one dropout curve per sample, because the number of missing proteins can differ substantially between samples and the effect cannot be fixed by normalization. We find ρ_j and ζ_j as the parameters that maximize

$$\begin{aligned}\log p(\rho_j, \zeta_j | Y, \hat{\boldsymbol{\mu}}_j, \boldsymbol{\sigma}^2_{\hat{\boldsymbol{\mu}}_j}) \propto & \sum_{i:y_{ij}\neq \texttt{NA}} \log\left(1 - \Phi(y_{ij}; \rho_j, \zeta_j^2)\right) \\ & + \sum_{i:y_{ij}=\texttt{NA}} \log\left(\Phi(\hat{\mu}_{ij}; \rho_j, \zeta_j^2 + \sigma^2_{\hat{\mu}_{ij}})\right),\end{aligned} \tag{A28}$$

where we use the predicted values $\hat{\mu}_{ij}$ for the missing observations and the associated uncertainty

$$\sigma^2_{\hat{\mu}_{ij}} = X_j' \Sigma_i X_j. \tag{A29}$$

We use the general purpose optimizer implemented by the R function `optim` (Broyden 1970, Fletcher 1970, Goldfarb 1970, Shanno 1970) to find the maximum.

b. Variance Prior

In the model without missing values, Smyth (2004) has described how to estimate the hyper-parameters of the variance prior τ_0^2 and df_0 from the unbiased variances s_i^2 and the degrees of freedom. In the previous section (Equation (A20) and (A21)), we have shown how derive those values in the case of missing values. But if we were to use those values directly we would have the problem that they already contain the information of last rounds hyper-parameter, and thus the variance prior would get narrower and narrower. To avoid this problem, we recalculate the quantities from the last section without location and variance moderation (simply by setting the first two lines of Equation (A10) to 0) and call them ${}_u\hat{s}^2$ and ${}_u\widehat{\mathrm{df}}$. We use those for the inference of τ_0^2 and df_0, which are just the quantities that maximize the log likelihood

$$\sum_i \log f_F\left({}_u\hat{s}_i^2; \tau^2 = \tau_0^2, \mathrm{df}_1 = {}_u\widehat{\mathrm{df}}_i, \mathrm{df}_2 = \mathrm{df}_0\right). \tag{A30}$$

c. Location Prior

Lastly, we will explain how to find the hyper-parameters for the prior on the protein means. Equation (A3) states that we believe that the proteins means μ_i are drawn from a Normal distribution. We estimate the mean of that location prior using a trimmed mean of the predicted values across all proteins and samples

$$\mu_0 = \text{trimmed-mean}_{0.2}(\hat{\mu}_{ij}). \tag{A31}$$

But, we cannot calculate the variance the same way, because using the already regularized values $\hat{\mu}_{ij}$ would lead to narrower and narrower estimates. This means that we need the un-regularized value ${}_u\hat{\mu}_i$.

We are only able to calculate ${}_u\hat{\mu}_i$ if we have at least one observation, but we more likely to have proteins without any observations left of the global mean μ_0. Thus, we will ignore all ${}_u\hat{\mu}_{ij} < \mu_0$ and assume that the distribution is symmetric.

To find the empirical Bayesian estimate of σ_0^2, we use the approach described by Efron and Morris (1975) for a Normal prior density, who showed that σ_0^2 is the value that solves

$$\sigma_0^2 = \frac{\sum_{i,j}({}_u\hat{\mu}_{ij}{}^2 - \sigma^2_{{}_u\hat{\mu}_{ij}})(\sigma_0^2 + \sigma^2_{{}_u\hat{\mu}_{ij}})^{-2}}{\sum_{i,j}(\sigma_0^2 + \sigma^2_{{}_u\hat{\mu}_{ij}})^{-2}} \tag{A32}$$

which we find using the `root` function in R.

Appendix B: Distances

Understanding which samples are similar and which are not is an important step for quality control. Again, missing values make this task difficult. If we were to impute the missing values, we would get unrealistic high similarity for samples with many missing values. Instead, we propose to construct a probabilistic similarity measure.

The most typical measure of sample similarity is the Euclidean distance between two samples in the feature space. If there were no missing values this would just be

$$\text{dist}_{1,2} = \sqrt{\sum_i (y_{i1} - y_{i2})^2}. \tag{B1}$$

The feature space is the I dimensional space where each axis corresponds to one protein and each sample is a point in that space. If a protein measurement is missing, we know that its intensity was low, but we cannot exactly say where along that particular axis the point is. Thus, we will convert the deterministic point $\boldsymbol{x}_{\cdot j}$ into a multivariate Gaussian with a diagonal covariance matrix Σ. The entries on the diagonal of Σ are zero if the protein was observed and correspond to the uncertainty of the protein intensity if it is missing. Equivalently, the mean vector of the Gaussian $\boldsymbol{\mu}$ is either the intensity measurement for observed proteins or the mean of our estimate where a missing value could realistically have been. As we are not certain anymore where each sample is in the feature space, we cannot calculate a deterministic distance between two samples, but we can estimate the expected distance and the associated uncertainty. Mathai and Provost (1992, p.53) provide exact formulas for the moments for the squared distance

$$\begin{aligned}\mathbb{E}[\text{dist}^2] = &\sum_i (\mu_{i1} - \mu_{i2})^2 \\ &+ \sum_i (\sigma^2_{\mu_{i1}} + \sigma^2_{\mu i2})\end{aligned} \tag{B2}$$

and

$$\begin{aligned}\mathbb{V}[\text{dist}^2] =& 4\sum_i (\mu_{i1} - \mu_{i2})^2 \left(\sigma^2_{\mu_{i1}} + \sigma^2_{\mu i2}\right) \\ &+ 2\sum_i \left(\sigma^2_{\mu_{i1}} + \sigma^2_{\mu i2}\right)^2.\end{aligned} \tag{B3}$$

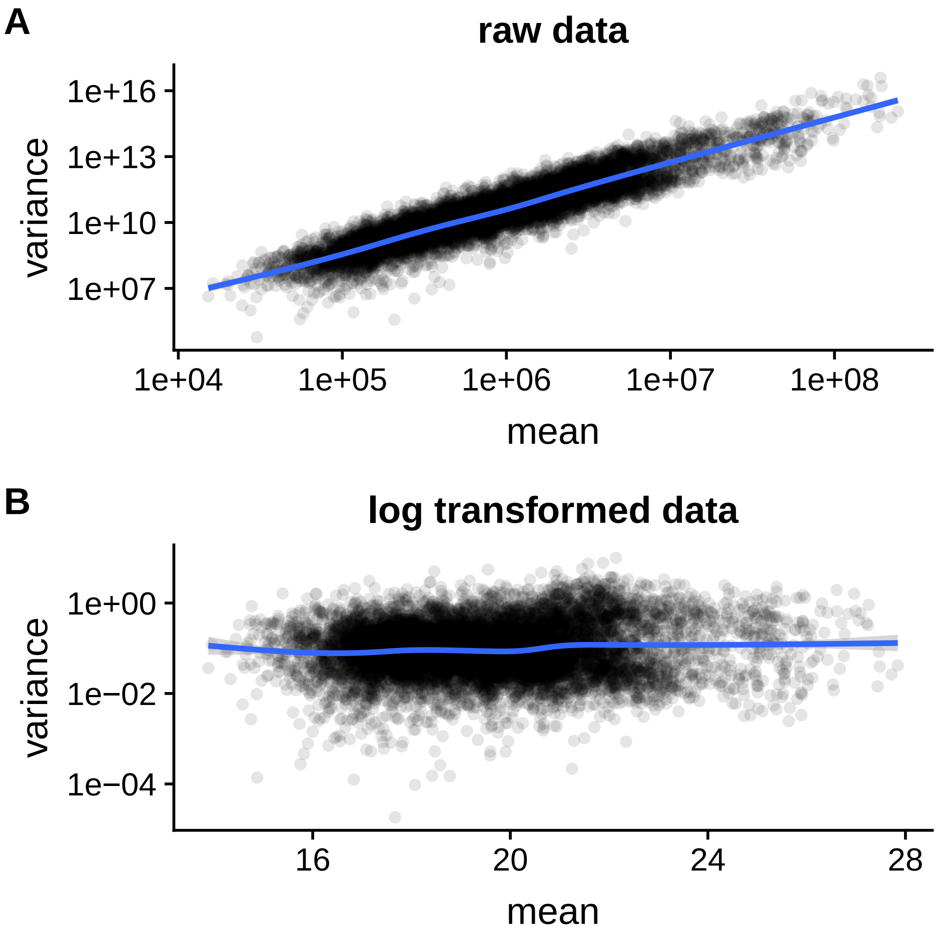
We use those equations to approximate the actual quantities of interested the estimated distance and the associated uncertainty

$$\mathbb{E}[\text{dist}] \approx \sqrt{\mathbb{E}[\text{dist}^2]} \tag{B4}$$

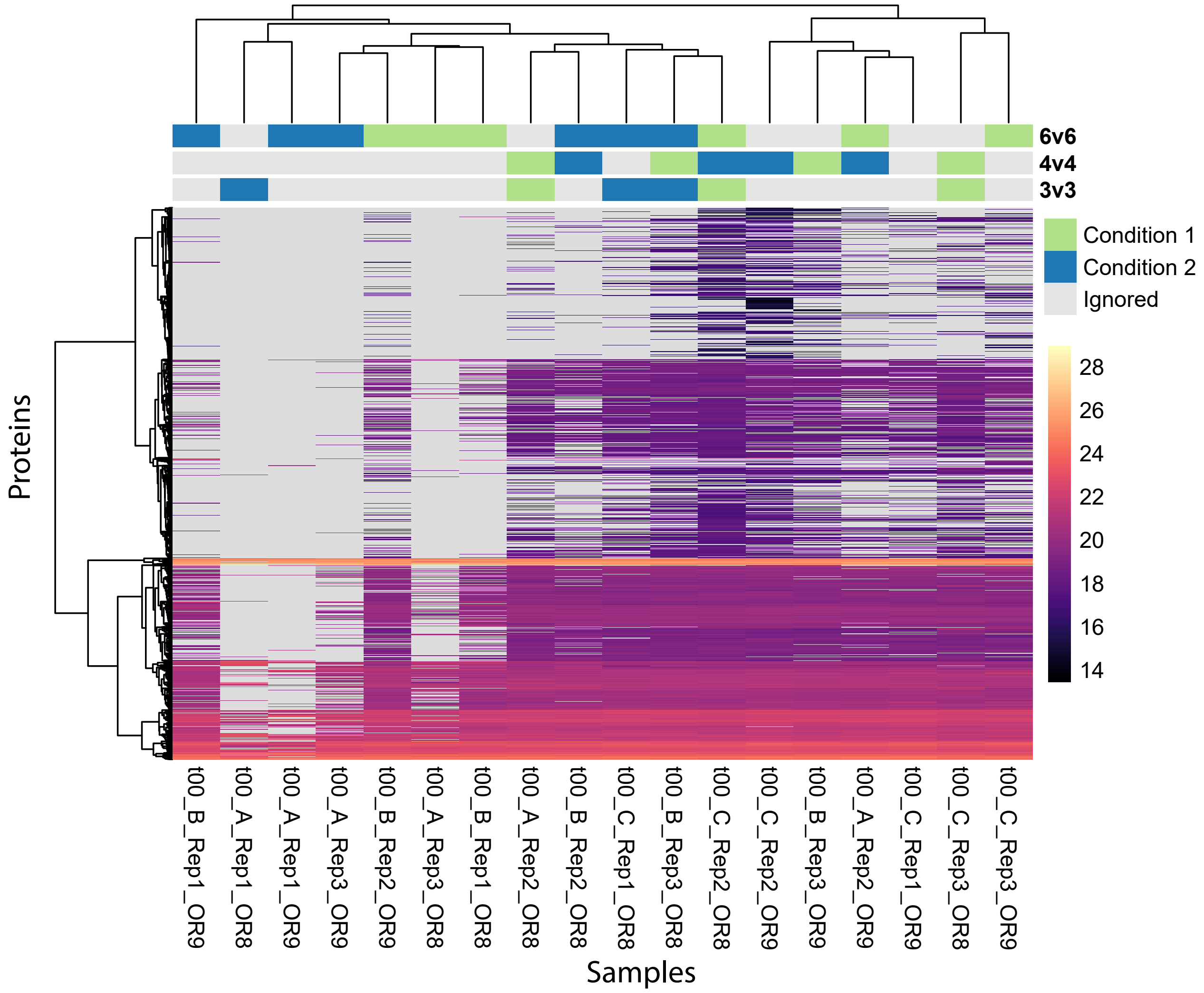
and

$$\begin{aligned}\mathbb{V}[\text{dist}] &\approx \mathbb{V}[\text{dist}^2] \left(\left. \frac{d\sqrt{x}}{dx} \right|_{x=\mathbb{E}[\text{dist}^2]} \right)^2 \\ &\approx \frac{\mathbb{V}[\text{dist}^2]}{4\mathbb{E}[\text{dist}^2]}.\end{aligned} \tag{B5}$$

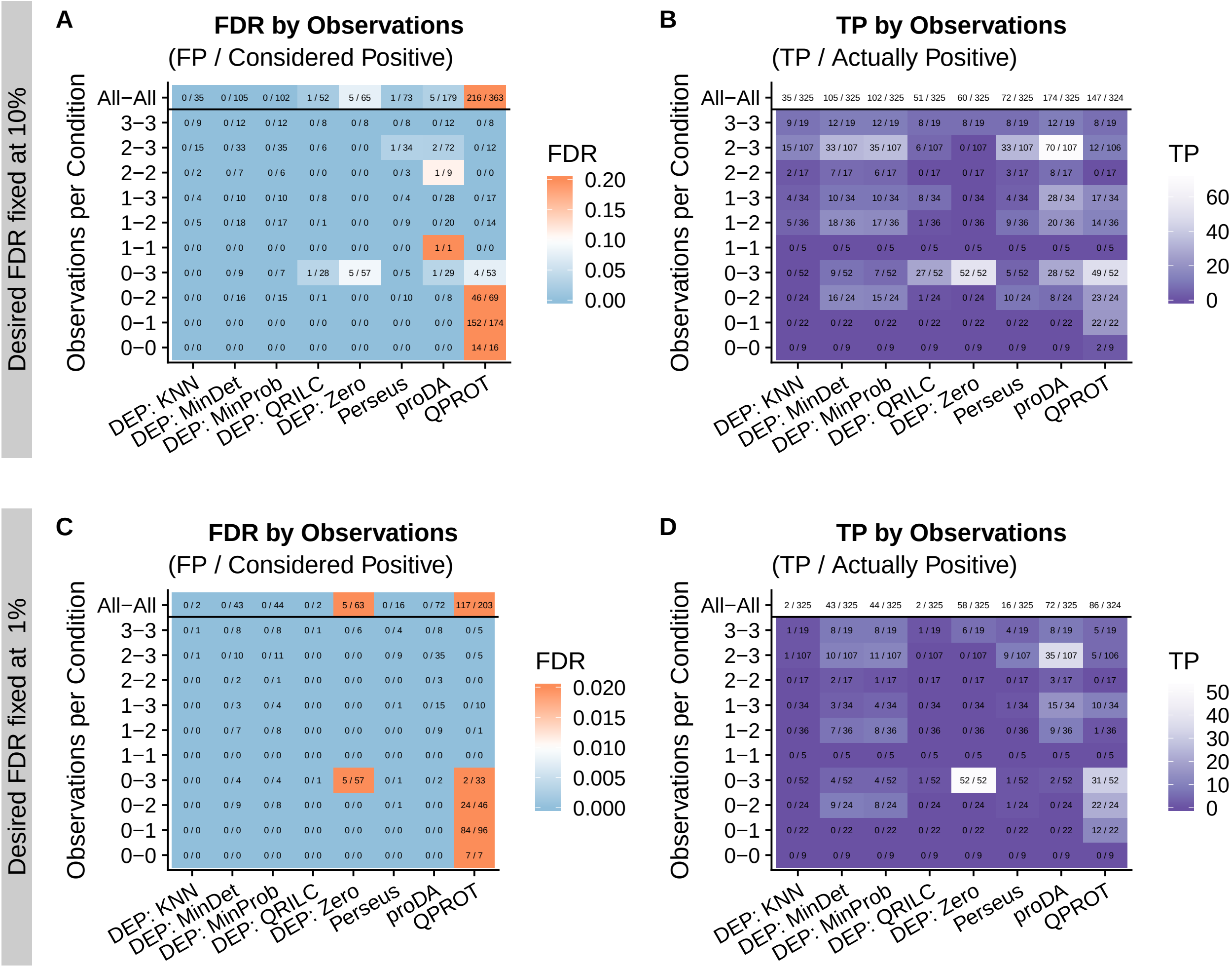
Appendix C: Supplementary Figures



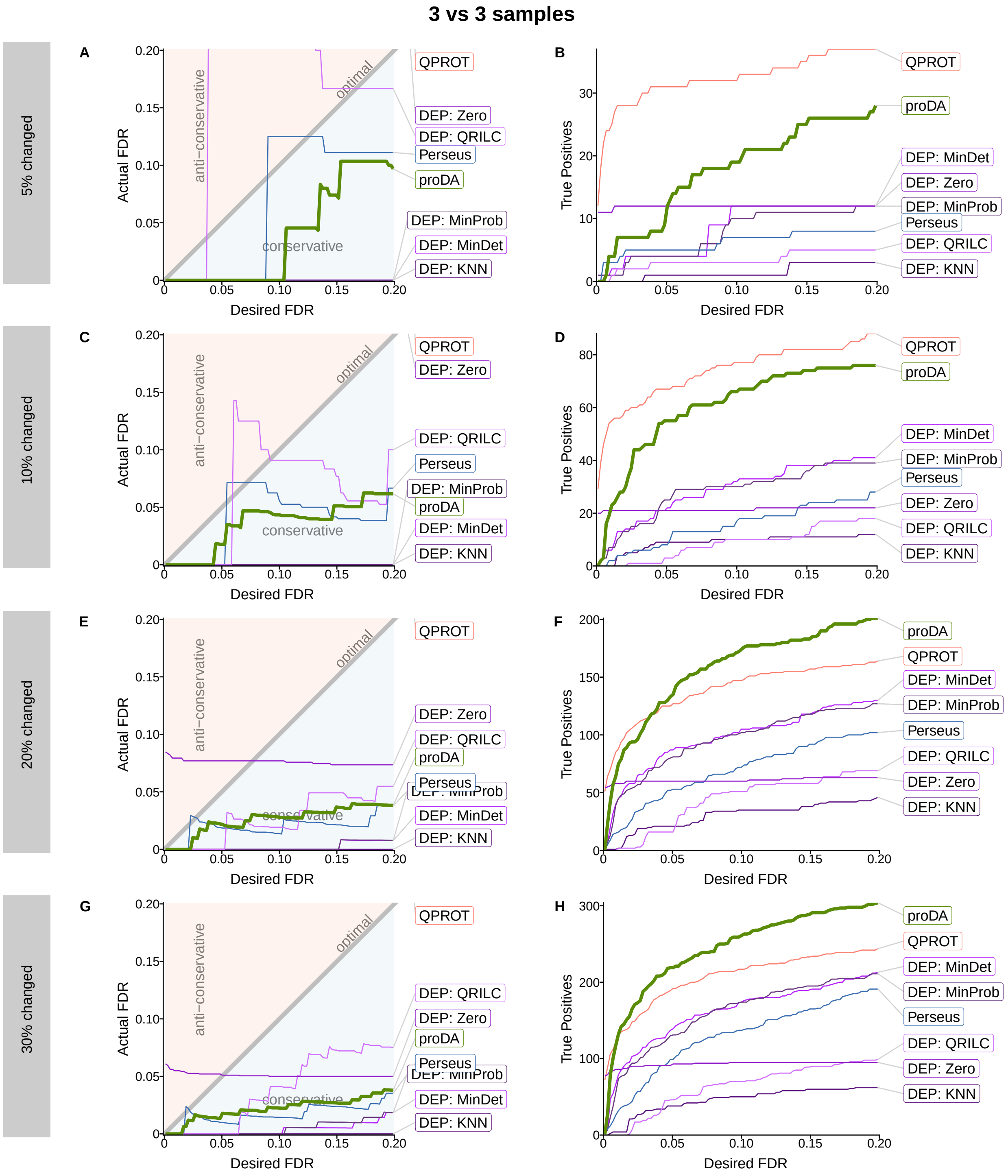
Suppl. Figure S1: Mean-variance relationship on the full de Graaf dataset. Each dot represents mean and variance for one protein at one time point and MS run. The blue line is a ggplot2 smoothing fit. A) the mean-variance relation on the raw data. B) the mean variance relation on the $\log_2$ transformed data.



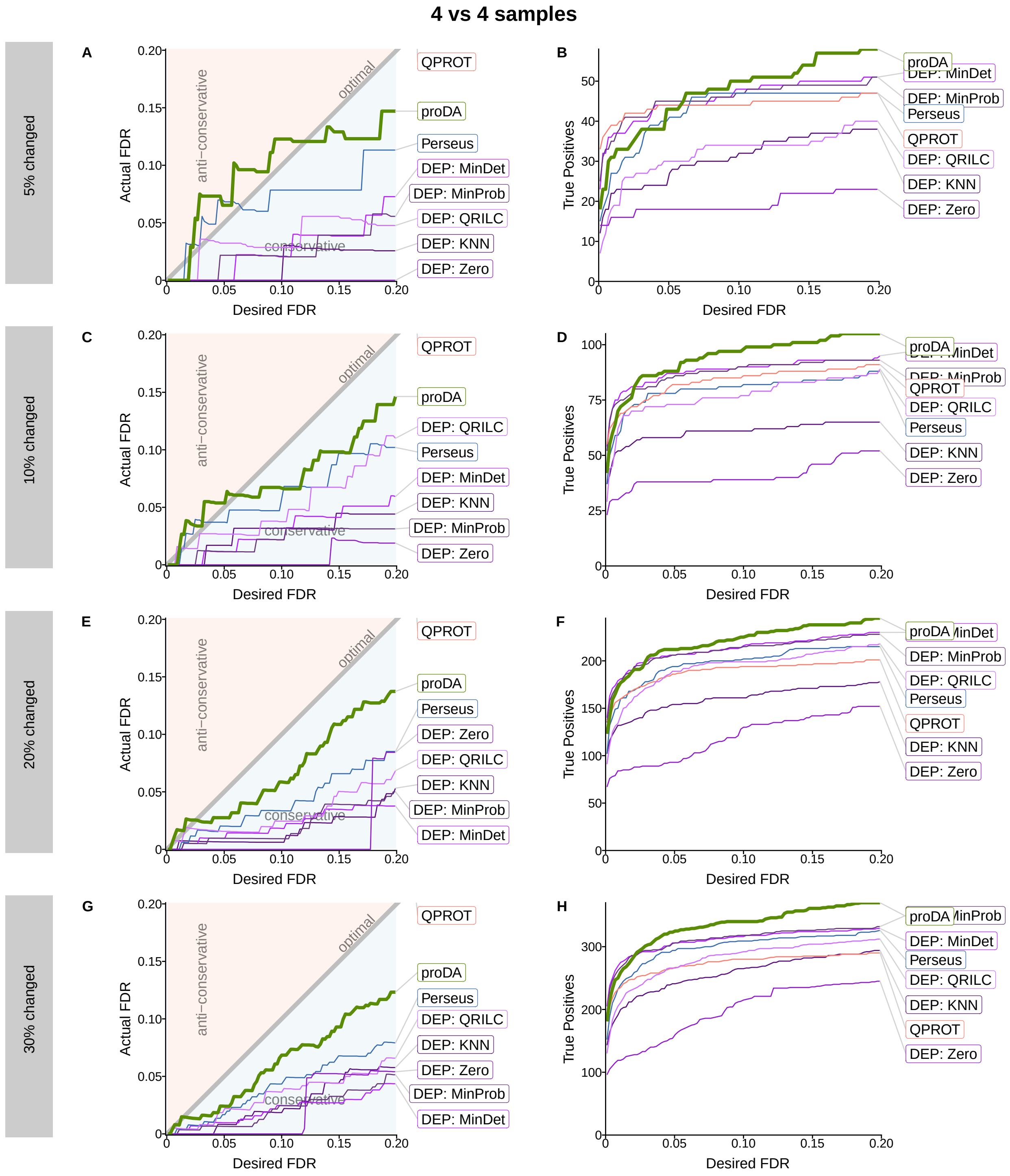
Suppl. Figure S2: Heatmap of the de Graaf dataset. Each column is a sample and each row is a protein. The color shows the intensity of the respective protein and missing values are greyed out. The samples and proteins are clustered using a hierarchical clustering on the expected distances calculated with Equation (B4). The annotations on top of the heatmap indicate which samples were compared in the different performance and calibration experiments shown in Figure 3 and Supplementary Figure S4-S6.



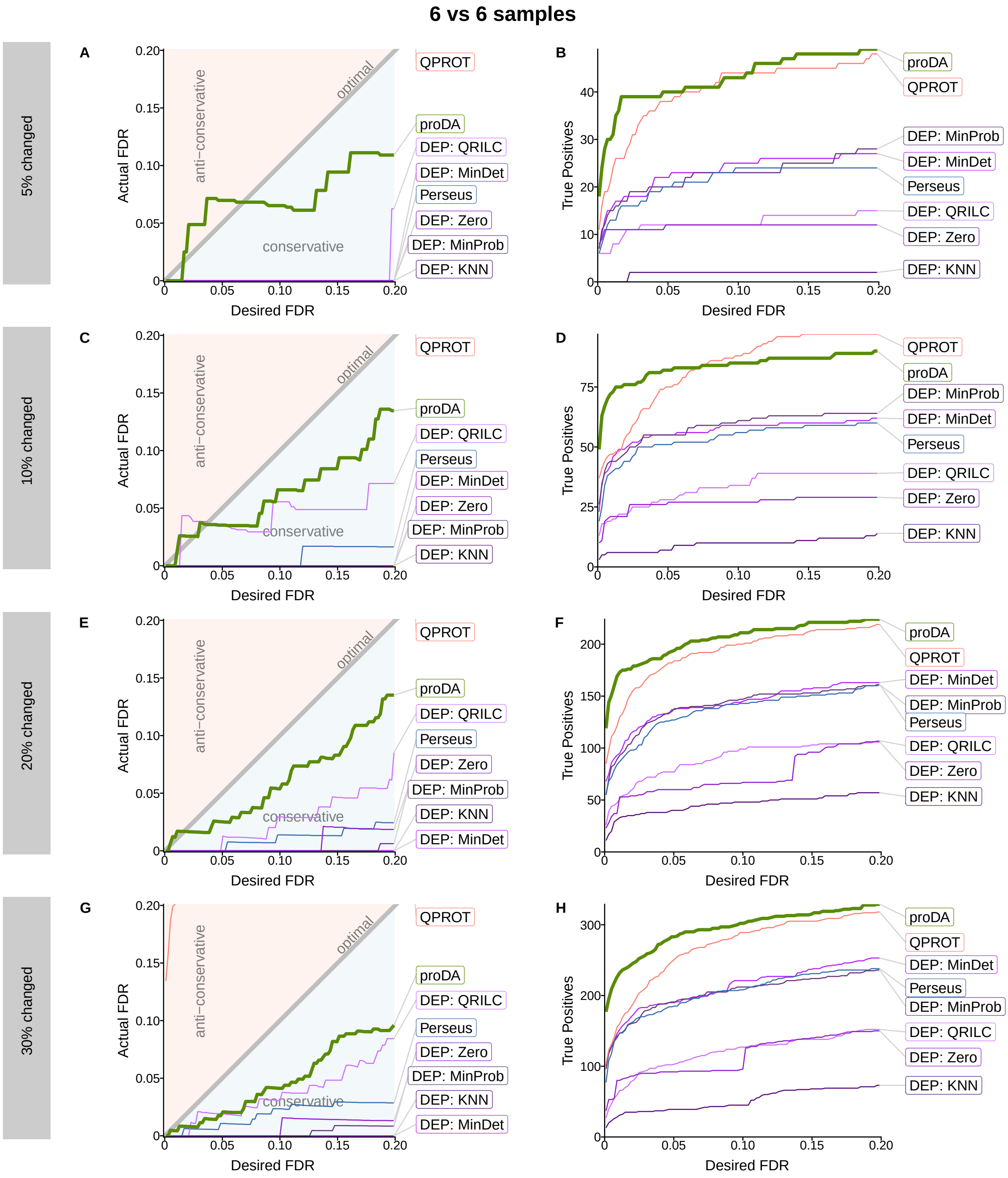
Suppl. Figure S3: Calibration and performance stratified by the number of observations on the three vs three comparison on the de Graaf dataset with 20% changed proteins. The x-axis shows the four tested methods and the y-axis the number of observed values in condition one and two. The smaller number is listed first. In addition the top line on the y-axis shows the marginal over all combination of observations. A) and B) show the results when fixing the desired FDR at 10%. A) shows the actual FDR for each method and B) shows the number true positives. C) and D) show the same features if the desired FDR is fixed at 1%. The color scales show the FDR (A and C) or the number of true positives (B and D). White indicates the optimal value. In A) and C) light blue color indicates a conservative FDR, whereas orange indicates an anti-conservative FDR.



Suppl. Figure S4: Calibration and performance comparison with three vs. three samples. A,C,E,G) comparison of the desired FDR with the FDR that is actually produced by the tool acording to the ground truth. The line for the QPROT method is missing because it is literally of the charts. B,D,F,H) Plot of how many actually changed proteins (true positives) each method identified at a specified FDR level.



Suppl. Figure S5: Same as Supplementary Figure S4, but for a comparison of 4 vs 4 samples.



Suppl. Figure S6: Same as Supplementary Figure S4, but for a comparison of 6 vs 6 samples.