Imputation of label-free quantitative mass spectrometry-based proteomics data using self supervised deep learning

Authors

Henry Webel[1], Lili Niu[1], Annelaura Bach Nielsen[1], Marie Locard-Paulet[1], Matthias Mann[1,3], Lars Juhl Jensen[1], Simon Rasmussen[1,2]

Affiliations

[1] Novo Nordisk Foundation Center for Protein Research, Faculty of Health and Medical Sciences, University of Copenhagen, Copenhagen N, Denmark

[2] The Novo Nordisk Foundation Center for Genomic Mechanisms of Disease, Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA

[3] Department of Proteomics and Signal Transduction, Max Planck Institute of Biochemistry, Martinsried, Germany

Keywords

Imputation, self-supervised learning, deep learning, reproducibility, mass spectrometry, proteomics

Abstract

Imputation techniques provide means to replace missing measurements with a value and are used in almost all downstream analysis of mass spectrometry (MS) based proteomics data using label-free quantification (LFQ). Here we demonstrate how collaborative filtering, denoising autoencoders, and variational autoencoders can impute missing values in the context of LFQ at different levels. We applied our method, proteomics imputation modeling mass spectrometry (PIMMS), to an alcohol-related liver disease (ALD) cohort with blood plasma proteomics data available for 358 individuals. Removing 20% of the intensities we were able to recover 20 out of 22 significant abundant protein groups using PIMMS-VAE imputations. When analyzing the full dataset we identified 27 additional proteins (+11.8%) that were significantly differentially abundant across disease stages compared to no imputation and found that some of these were predictive of ALD progression in machine learning models. We, therefore, suggest the use of deep learning approaches for imputing missing values in MS-based proteomics and provide workflows for these.

Introduction

Proteomics is a technology for the identification and quantification of proteins to answer a broad set of biological questions[1] and together with RNA and DNA sequencing offers a way to map the composition of biological systems. It is widely applied across many fields of research including identification of biomarkers and drug targets for diseases such as alcoholic liver disease (ALD)[2], ovarian cancer[3] and Alzheimer's disease[4]. Different workflows have been developed for analysis of body fluids, cells, frozen tissues and tissue slides, and are rapidly evolving. Recent technological advancements have enabled proteome analysis at the single cell or single cell-population level[5,6], allowing the selection of single cells using image recognition[7]. However, for most approaches, missing values are abundant due to the semi-stochastic nature of precursor selection for fragmentation and need to be replaced for at least some parts of the data analysis. Currently, imputation of missing values in proteomics data usually assumes that the protein abundance was below the instrument detection limit or the protein was absent. In general, the community differentiates between missing at random (MAR) which is assumed to affect all intensities across the dynamic range, whereas missing not at random (MNAR) becomes more prevalent the more the intensity of a peptide approaches the limit of detection of the instrument. However, not all missing values are due to this mechanism, and by assuming the limit of

detection as the reason for missingness will lead to potentially wrong imputations and subsequently to biased statistical results that are limiting the conclusion from data. Various acquisition methods have been developed including data-independent acquisition (DIA), BoxCar and PASEF to alleviate the “missing value” problem in data-dependent acquisition (DDA) methods[8–10]. Advances in informatics solutions have also greatly improved data analysis of mass spectra acquired by these acquisition methods and consequently proteome depth and data completeness[11]. However, missing value imputation of search results for downstream analysis remains a recurring task for most applications. The noise in data from the instrument as well as peptide identification is most abundant for label-free quantification proteomics in DDA with missingness ranging from 10-40%[12], but for instance, blood plasma measured using DIA in a study of ALD still contained 37% missing values across all samples and protein groups before any filtering. Independent of the proteomics setup, once data is to be analyzed, the remaining missing values between samples have to be imputed for most methods. Therefore, how they are handled will influence the downstream results.

Several methods have been evaluated for imputation of MS proteomics data with an overview and a benchmark provided in work by Wang et al., using NAGuideR[13]. Despite an abundance of imputation methods, an often-used approach to impute data at the protein group level is by using random draws from a down-shifted normal (RSN) distribution. The mass spectrometry (MS) signal comes from ions and most people are interested in the summary of ions through peptide spectrum matches to groups of proteins. The protein intensities, stemming from aggregation of the precursor and/or fragment ion values in MS1 and MS2 scans, are assumed to be log-normally distributed, i.e. the log transformed intensities are entirely determined by their mean and variance. In RSN replacements are then drawn using a normal distribution with a mean shifted towards the lower detection limit with a reduced variance. This is done on the assumption that the data is left-censored, i.e. that proteins are missing due to absence or lower abundance than the instrument detection limit. Following this line of thought, several studies focus on determining what works best for different causes of missing values using some form of simulation[12,14]. Other studies focus their analysis on post-translational modifications[15], the best combination of software tools, datasets and imputation method[16], normalization and batch effects correction[17] or downstream analysis[18]. Other methods have been developed to handle specific missing mechanisms, for instance, random imputation, fixed value imputation such as limit of detection or x-quantile of feature, model-based imputation using k-nearest neighbor (KNN), linear models[12] or tree-based models[13]. These either impute using a global minimum, a

statistic calculated on a single feature or a few features, with the need to iteratively consider each feature at a time. Finally, approaches such as DAPAR and Prostar offer several methods for imputing left-censored data, e.g. the widely used drawing from a normal distribution around the lower detection limit where the Gaussian mean and variance are estimated using quantile regression, abbreviated QRILC[14,19,20].

Most previous work on developing methods for imputation of MS proteomics data focus on small scale setups where they for instance evaluate two separate groups in three replicates[16,21]. Other studies prefer to reduce the number of initial missing values by transferring identification from one run to the next using e.g. Match Between Run implemented in MaxQuant or cross-assignment by Proline[22,23]. Alternatively, an established laboratory method to tackle variability in small-scale setups is to use replicates of samples to have complementary measurements, in order to transfer identifications between runs[24]. The specific evaluation strategy varies between the setup of the data and the missing values simulation approach, but to our knowledge, no scalable workflows are provided to run the evaluation on a new dataset generating a validation and test dataset, as well as a generic and flexible approach to imputation.

We turn to machine learning for imputation of missing intensities as it offers the possibility to learn from the data itself. Deep Learning (DL) has been used to improve over existing machine learning models in a variety of biological data problems[25–27]. DL has been successfully applied to predict peptide features such as retention times, collisional cross-sections and tandem mass spectra, significantly boosting the peptide identifications and precision of searches of MS-based proteomics[28–32]. We use intensities from search results generated from tools like e.g. MaxQuant or Spectronaut and apply DL methods to impute these. We considered three types of models that process the search results of precursors, peptides or protein groups slightly differently. First, we considered a collaborative filtering (CF) approach, where each feature and each sample is assigned a trainable embedding. Second, we considered an autoencoder with a deterministic latent representation - a denoising autoencoder (DAE). Third, we considered a variational autoencoder (VAE) as a generative model that encodes a stochastic latent representation, i.e. a high-dimensional Gaussian distribution. Although the inputs to all models are intensities for a feature and sample combination, the training objectives, complexity, and therefore capabilities of the models are different. The CF and autoencoder objective only focuses on reconstruction, whereas the VAE adds a constraint on the latent representation.

Furthermore, the first two modeling approaches use a mean-squared error (MSE) reconstruction loss, whereas the VAE uses a probabilistic loss to assess the reconstruction error.

Here we used large (N≈450) and smaller (N≈50) MS-based proteomics datasets of HeLa cell line tryptic lysates acquired on a single machine (Q Executive HF-X Orbitrap) over a period of roughly two years to evaluate the general performance on repeated measurements of an homogenous biological sample on a medium to large-scale dataset. We applied the DL models described above (CF, DAE, VAE) which create feature representation holistic for the entire distribution in a given dataset prior to any normalization. For evaluation, we developed a workflow that allowed comparison between the three DL models and other approaches listed in the NAGuideR tool (**Supp. Table 1**). Finally, we applied the VAE to a study of ALD patients and found 27 (+11.8%) more significantly differential abundant protein groups in comparison to no imputation and showed that additional protein groups could be leveraged for predicting disease. We termed our set of models and workflows proteomics imputation modeling mass spectrometry (PIMMS) and made the workflows, code, and example configs available at https://github.com/RasmussenLab/pimms. To enable reproducibility and adaptation to new data and strategies, we share our Python code along snakemake workflows.

Results

Evaluating self-supervised models for imputation of MS data

We assessed the capability of three unsupervised models for proteomics data imputation. First, we considered modeling proteomics data using CF assigning each sample and each feature an embedding vector and using their combination to predict intensity values. Second, we considered a standard autoencoder, training it using a denoising strategy that has to learn to reconstruct masked values making it a DAE. Third, we applied a VAE with a stochastic latent space (**Fig. 1a**). The two autoencoder architectures used all data to represent a sample in a low dimensional space, which was used to reconstruct the original data. In contrast, the CF model had to learn both a latent embedding space for the samples and features. We compared these to a heuristic-based approach of median imputation per feature across samples and other machine learning approaches such as k-nearest neighbors (KNN) or random forest (RF). While the DL methods, KNN and median imputation are able to also impute large datasets, most reference methods implemented in R failed to scale to high-dimensional data.

Our development dataset consisted of 564 HeLa runs of one Q Executive HF-X Orbitrap generated during continuous quality control of the mass spectrometers. We initially investigated the structure of the dataset using the first two principal components (**Fig. 1b**), which grouped the samples into two separate clusters. The median prevalence per protein group, i.e. the median number of samples where a protein group was detected, was 526 samples [min: 45, max: 564], and samples had a median of 3,768 protein groups [min: 2,170, max: 4,185]. We generated another smaller development dataset of using the most recent 50 samples (**Fig. 1c**) to test dependence of performance on sample size. Whereas median imputation per feature across samples did not condition their imputation on the value of other features in a given sample, our and other machine learning approaches consider other feature values. Validation and test data were drawn according to the entire data distribution from all samples in a dataset.

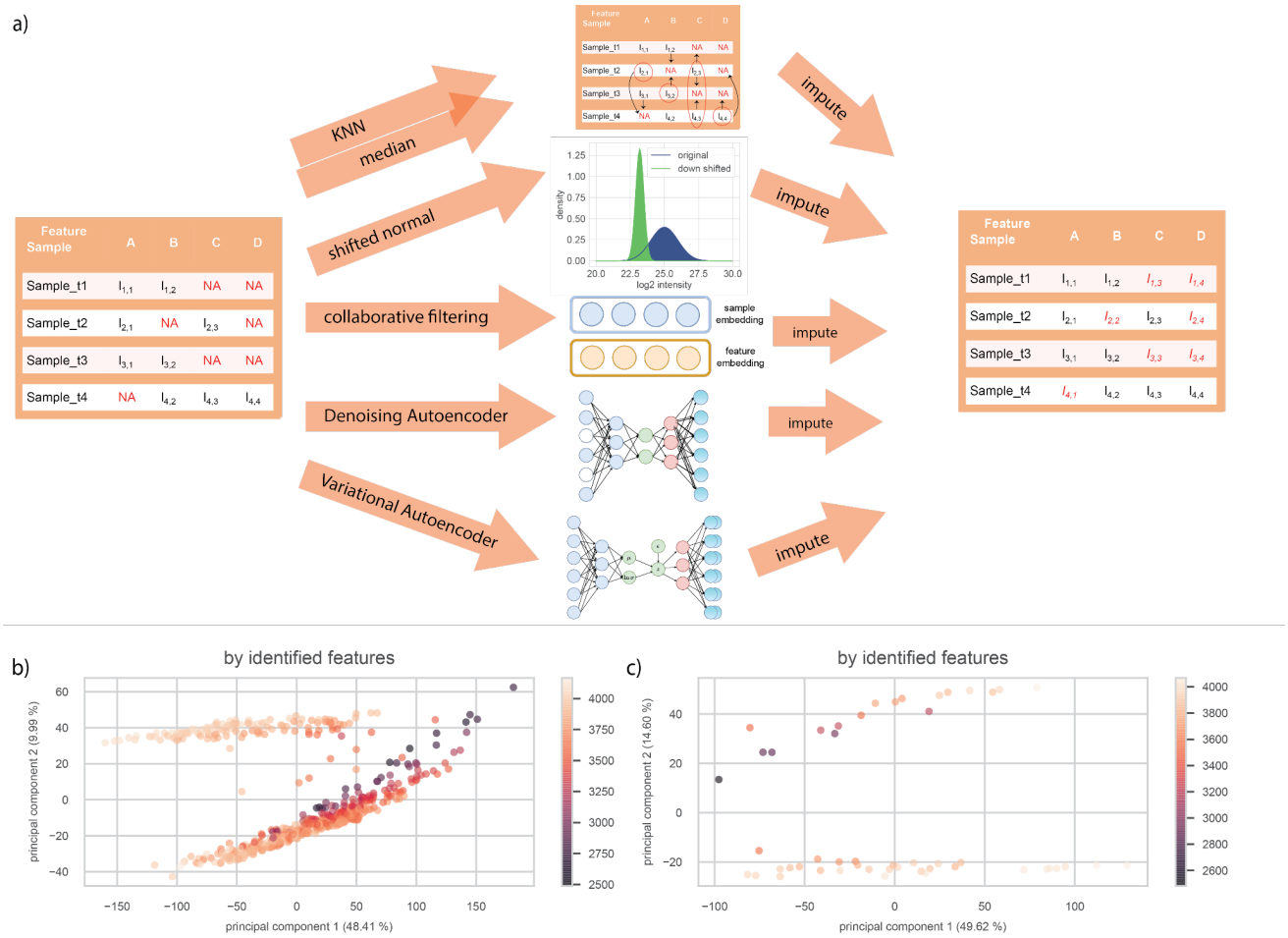

Figure 1: Overview of workflow for downstream analysis tasks and HeLa dataset. a) Results taken from MS data analysis software (search and quantification) were used as input for downstream analysis. Here we used MaxQuant for data dependent acquisition to analyze raw MS data. We compared three different self-supervised DL

approaches with methods mentioned in NAGuideR: median imputation and KNN interpolation exemplified. Random Forest did not scale to high dimensional data. **b) P**rincipal component one versus two of 539 selected HeLa runs for protein groups recorded on one instrument. **c)** Same as b, based on the 50 runs forming the small development dataset. We used a cutoff of 25% feature prevalence across samples to be included into the workflow shown in (a). Samples were filtered in a second step by their completeness of the selected features (**Fig. S1**).

Imputing precursors, aggregated peptides, and protein group data

We applied the imputation methods to the development dataset, e.g. consisting of 539 samples for protein groups using our selection criteria (**Supp. Fig. S1**). We ran several configurations using a grid search to find the best configurations using simulated missing values in a validation and test split from all samples (see Methods, **Supp. Data 1, 2**). In absolute numbers these were 100,001 for the protein group, 616,561 aggregated peptides and 661,817 precursors intensities in the test set for our development dataset of instrument 6070. When investigating the performance of the imputation methods we used the mean absolute error (MAE) on the log2 scaled intensities between predicted and true measured intensity values on our simulated missing values. We found the DL approaches to have less than half MAE compared (0.43, 0.41 and 0.44 for CF, DAE and VAE, respectively) to the median imputation MAE of 1.02. This was consistent across the entire distribution of protein group intensities. KNN of samples across HeLa cell line measurements had a MAE of 0.46 using the scikit learn implementation and of 0.51 using an R based implementation using impute[33,34]. The R based random forest (RF) imputation[35] only completed for the protein groups and had an MAE of 0.46. Therefore, the KNN or RF were slightly (1.07 times) worse in comparison to the self-supervised models. Comparing the performance between levels of data aggregation (**Fig. 2a,c, Supp. Data 3**), we found similar patterns as with the protein groups where the self-supervised models reduced the MAE by roughly 20% compared to the KNN interpolation. Furthermore, we found the overall performance to be the worst for the protein-level data, better for aggregated peptides, and best for precursors. This is in line with previous results of Lazar and co-workers that showed better performance for lower levels of aggregations as it avoids implicit imputations by a neutral element[14]. We omitted imputation by RSN as it was not able to model the entire data distribution (see Methods). Regarding execution times RF did not scale to the high dimensional datasets of peptides and precursors. Runtime for imputation scripts for protein groups including data loading, manipulations and training were between 24 seconds and 52 mins (KNN: 0:24, VAE: 0:53, DAE: 1:30, CF: 2:21, RF: 51:54) using a T2000 GPU with 4GB of memory for the PIMMS models (**Supp. Data. 4**). Runtimes of PIMMS models for the precursor were higher (Median: 1:34, KNN: 2:30, DAE: 4:40, VAE: 5:16 and CF: 16:12). In general runtimes varied based on the

199 number of epochs, mini batch-size, model architecture, patience for early stopping and the initial
200 random weights. Additionally, the grid search results showed that the models could be trained
201 without prior normalization and were able to fit the data using many different hyperparameter
202 configurations (**Supp. Fig. S2**). Reducing the training sample by roughly a tenth, performance
203 on the smaller development dataset (N=50) was comparable to the performance on the larger
204 development dataset, while runtime reduced by 5 to 15 times for precursors (Median: 0:23,
205 KNN: 0:33, VAE: 1:04, DAE: 0:52, CF: 1:06). For the smaller development dataset on the
206 protein groups level, additional R methods from NAguideR were able to complete. We found
207 that BPCA[36] performed best on the 50 homogenous HeLa samples with relatively long runtime
208 (e.g. on precursors: BPAC: 20:44, MICE-NORM: 4:42). For 10, 20, 30 and 40 samples, other
209 methods showed good results, although PIMMS models still fit the data (**Supp. Data 3**): For
210 example with 10 samples the precursors were fitted best with an MAE of 0.57 using BPCA and
211 0.61 using CF. In summary, this indicated that on an unnormalized, intensity varying dataset
212 from a single machine the PIMMS models were able to capture patterns between detected
213 features to impute values (**Supp. Fig. S3a**) for as few as 50 samples with competitive
214 performance to other state-of-the-art methods.

215 **Imputation was consistent across a wide range of intensities**

216 We evaluated the performance across the dynamic range of intensities by binning test
217 intensities by their feature's training median and reporting the average error per bin. For protein
218 groups on both development datasets the minimal median intensity was 23 for protein groups
219 (**Fig. 2c-d**). The average error for intensities was roughly twice as large for intensities from the
220 minimum median intensity bin in comparison to the overall performance (**Fig. 2e-f**).
221 Interestingly, worst performance was observed for protein groups with the highest observed
222 median intensity, which for intensities from one or six protein groups had a two to four times
223 worse MAE than the overall MAE. The relative performance between the models was consistent
224 across the bins. If we compared this to the relative distribution of imputed intensities (with
225 errors) to the unmodified intensities in the test split, we found that low intensity values were
226 partly imputed towards the center of the intensity distribution (**Fig. 2g**). Furthermore, we
227 investigated whether there was a difference in the accuracy of the imputation based on how
228 often a protein group was observed. Here we found that the MAE varied between 0.6 and 0.8
229 for proteins observed in 25-80% of the samples, whereas for proteins observed in more than
230 80% of the samples the MAE decreased to below 0.4 (**Fig. 2c, e**). We observed a similar trend
231 when analyzing the smaller dataset of only 50 samples (**Fig. 2d, f**). This indicated that some

protein groups were harder for all the methods to impute than others, but also that the CF, DAE, and VAE predicted consistently across a wide range of protein groups intensities. We found similar results for the two other levels of data, aggregated peptides and precursors (**Supp. Fig. S4**). Finally, we investigated how stable the self-supervised models trained. In a first step, we randomly permuted all protein groups of the large development dataset, which leaves the median performance unchanged. Accordingly, training models on randomly permuted data could not outperform the imputation by the median of the training data split with an MAE of 1.04 (**Supp. Fig. S5a, Supp. Data 5,** CF: 1.05, RF, VAE: 1.08, DAE: 1.12). Next, we trained the self-supervised models five times on the same data split as well repeated training five times on new splits. The performance between fitted models varied in a narrow margin (CF: 0.427-0.433, DAE: 0.424-0.436, VAE: 0.438-0.453 [Min-Max]) (**Supp. Fig. S5b,c, Supp. Data 5**). We therefore conclude that the DL models could be consistently fitted to data and that self-supervised models were able to fit the data holistically for imputation purposes.

Competitive within-sample and feature-wise between-sample correlation

We evaluated performance without a specific distance measure by evaluating Pearson correlations of simulated missing values to the truth (**Supp. Fig. S6, Supp. Data 6**). The mean Pearson correlation between samples for protein groups was a bit higher for all self-supervised models than for the RF and KNN imputations on the smaller development dataset (BPCA: 0.84, CF: 0.84, SEQKNN, VAE, DAE: 0.83, RF:0.82, KNN: 0.80). For features that had less than three observations in the test split, the between-sample correlation was removed. The correlation between features within a sample was higher in general, with a mean correlation of around 0.96 for all models seen among the best five along the sorting by the MAE (DAE, CF, VAE: 0.96, RF, KNN: 0.95). This showed that the ordering within a sample was better than the correlation of protein groups between samples as the overall abundance level of single protein groups vary across samples. Both correlation comparisons, therefore, indicated that the three self-supervised models were able to model the data well without prior normalization of the data.

Figure 2: Different levels of MS-based proteomics data can be imputed using self-supervised DL models. a) Performance of imputation methods at the level of protein groups, aggregated peptides, and precursors for MaxQuant outputs. Mean averaged error (MAE) is shown on the y-axis. Blue: median, yellow: interpolation, green: CF, red: DAE, purple: VAE, blue: KNN (scikit-learn), brown: random forest (missForest), orange: knnmethod (impute). The DL methods performed better or equally good in comparison to other models. Performance was similar for all three

267 models on each data level and less aggregated data, i.e precursors and aggregated peptides perform better
268 compared to protein groups. RF did not finish for aggregated peptides and precursors. Models were ordered by
269 overall performance on the three datasets. **b)** As (a) but showing a decrease in performance for a subset of the data
270 with a maximum of 50 samples. The CF adapted better to smaller sample sizes. **c,d)** Protein groups medians were
271 binned by their integer median value and the boxplot of the percentage of missing values is shown for large and small
272 development. **e)** MAE for protein groups intensities in the test split of the larger development dataset binned by the
273 integer value of the protein group's median intensity in the training data split (4,403 protein groups, 100,101
274 intensities, models ordered by overall performance). Protein groups with a median intensity above 26 perform in the
275 range of the overall performance shown in a). **f)** MAE for protein groups in test split of smaller development dataset
276 binned by integer value of intensity (4,405 protein groups, 9,328 intensities, models ordered by overall performance).
277 **g**) Imputed and ground truth (observed) intensities for test data on large development dataset for top four overall
278 models.

279 **Performance of PIMMS on simulated missing values in real use case**

280 To assess the impact of imputation on a large real-world DIA dataset, we applied PIMMS to 455
281 blood plasma proteomics samples from a cohort of ALD patients and healthy controls[2]. After
282 imputation we again compared how well PIMMS imputed simulated missing values in the ALD
283 data and found that DAE, VAE, RF and CF achieved similar results of MAE between 0.43-0.45.
284 KNN was able to impute the data well, although slightly worse with MAE of 0.5 (**Fig. 3a, Supp.**
285 **Data 3**). The overall performance was nearly the same if we randomly removed 20% of the data
286 on a proportionally smaller test dataset (**Fig. 3b).** This can be explained by the relatively stable
287 measurements of the dataset (**Supp. Fig. S3b**) and indicates that most protein groups were
288 stable across patients as the simulated missing values were from the entire distribution of the
289 data. BPCA which performed well on the development dataset, was worse with an MAE of 2.4
290 and took more than 18 minutes to complete (**Supp. Data 3, 4**). As drawing random
291 replacements from a shifted normal distribution was used in the original study, we add this to
292 our comparison here. However, both the RSN and the interpolation were not really informative
293 for the comparison on the simulated data as basic assumptions are not met: RSN assumes
294 MNAR values and the simulated values were representative of the entire data, and therefore not
295 all low abundant. Similarly, the data does not contain replicates of the same sample, so the
296 interpolation strategy using KNN can only use the most similar samples. In the original study
297 RSN was used for imputation on a per sample basis, i.e. using the mean and standard deviation
298 of all protein groups in a sample. This yielded 12 times worse results on the simulated data
299 which, as mentioned above, did not meet the basic assumption of RSN (**Supp. Data 7)**.
300 Similarly, the correlation within samples and between samples was best for the self-supervised
301 models (**Supp. Data. 7**). The correlation across samples for each protein group (**Supp. Data 7**)

302 was lower for the ALD data in comparison to the heterogeneous measurements of HeLa cell line
303 data for the self-supervised models (median; DAE: 0.59, VAE: 0.55, RF: 0.56, CF: 0.52, KNN:
304 0.37) based on 355 protein groups with at least 3 intensities in test data split. The median
305 correlation within samples (**Supp. Data 7**) was around 0.97 for all models keeping the order by
306 MAE. This, thus, matched the overall results from the HeLa data analysis.

307 **Recovering lost differentially abundant protein groups when missing at random**

308 Then we tested the ability of the methods to recover differentially abundant protein groups when
309 removing data at random (MAR). Therefore, we randomly removed 20 percent of the data.
310 Comparing the results of the differential abundance analysis using the complete dataset
311 compared to the reduced dataset we observed a decrease in the number of differentially
312 abundant protein groups from 212 to 192 (ANCOVA, Benjamini-Hochberg multiple testing
313 correction, p-value≤0.05). Interestingly, we found that besides 22 protein groups losing their
314 differential abundance, four became differentially abundant by randomly removing data. To
315 assess the effectiveness of various imputation methods, we used no imputation as a
316 benchmark. We found that DAE, VAE, and RF imputation techniques successfully recovered 20
317 out of the 22 protein groups that lost their differential abundance. In contrast, median imputation
318 only recovered two protein groups, while CF imputation recovered 16 groups with three protein
319 groups slightly above the cutoff of 0.05. KNN recovered 13 protein groups (**Fig. 3c,e, Supp.**
320 **Data 8**). Four protein groups became differentially abundant without imputation when intensities
321 were removed in the 80% dataset, which we labeled as false positives to follow our approach
322 (**Fig. 3d,f**). However, three of the four were nearly statistically significantly abundant in the
323 100% dataset and therefore adding back patients through imputation to the analysis can be
324 seen as beneficial. This indicated that the PIMMS models were capable of recovering lost
325 signals of at least MAR data.
326

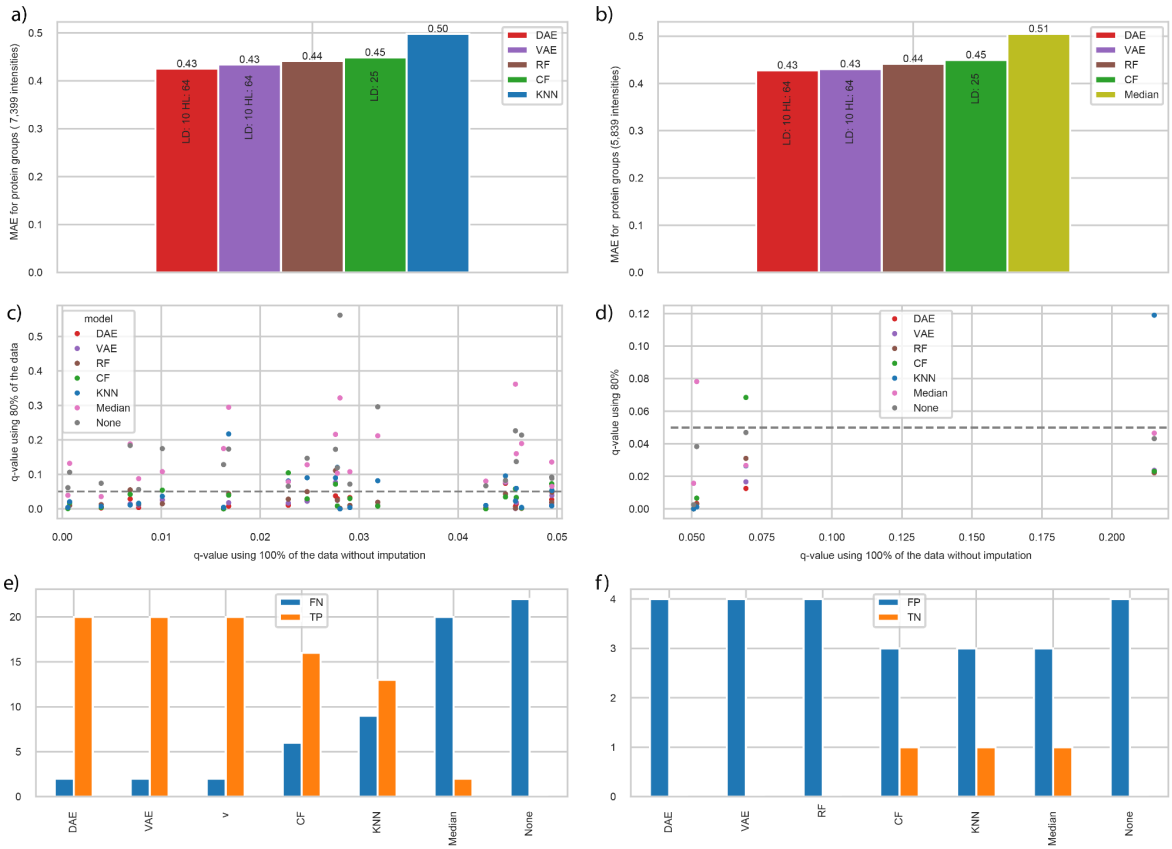

Fig. 3: Performance on simulated missing data in ALD dataset and effects of imputation on differential analysis. a) Performance for protein groups of plasma proteome data on simulated missing values on the full dataset and **b)** 80% dataset using the same configuration (LD: latent dimension, HL: hidden layer dimension). **c)** q-values for 22 protein groups which were differentially abundant using the full dataset without imputation, but not for 80 percent of the data without imputation. **d)** q-values for four protein groups which were differentially abundant using the 80 dataset without imputation, but not initially when using all data. **e)** Taking 22 the differentially abundant protein groups from c) on the full dataset as ground truth, count the false negative (FN) and true positives (TP) per method on the reduced dataset. **f)** Same as in e) for four protein groups in d), labeling differentially abundant as false positives (FP) since they did not show up as differentially abundant in the full dataset. True negatives (TN) are non differentially abundant here (**Supp. Data. 5**).

More differentially abundant proteins when using PIMMS

Then we investigated the number of differentially abundant protein groups as well as the ability of the plasma proteome to predict the fibrosis status of 358 individuals for the top five models and median imputation (**Supp. Data 9**). The fibrosis status was based on a liver biopsy, and individuals with all fibrosis scores ranging from zero to four were included. By using these models for imputation of missing data, leaving the available data as it was, we were able to perform the analysis for 377 protein groups (ANCOVA, Benjamini-Hochberg multiple testing correction, p-value≤0.05) compared to only 313 protein groups when using the RSN imputation approach as originally applied in Niu et. al (2022)[2]. Comparing RSN and VAE we observed that

347 both methods replaced missing values with a distribution shifted towards the lower abundance
348 region. However, the maximum intensity for missing values was higher for the VAE with a value
349 of 21 compared to the RSN with a value of 15 for protein groups (**Supp. Fig. S7**). Translating
350 the overall shift in distribution by the VAE to the RSN idea, the intensity distribution was shifted
351 by one standard deviation and the variance was shrunk by 0.7 in comparison to 1.6 standard
352 deviations and 0.3 shrinkage in the ALD-RSN setup. This difference underlies a fundamental
353 difference in the approaches and can be generalized for all other models (**Supp. Fig. S7**).
354 Whereas the RSN always assumes missing values due to low abundance, the VAE and other
355 approaches assign some missing values a higher intensity if other protein groups in the same
356 sample suggest that the missing value occurred rather due to a missed detection than low
357 abundance.
358
359 When performing differential analysis we found that 210 of the 313 protein groups originally
360 included were significantly differentially expressed using the RSN approach. However, 212 were
361 significant without any imputation and up to 220 were significant using the other methods (DAE:
362 220, VAE: 219, RF: 218, CF: 219, KNN: 217, Median: 217). Adding the newly included 64
363 protein groups, 255 of 377 in the PIMMS-VAE setup were differentially expressed, an increase
364 by 20% to the originally included and imputed protein groups with RSN (see Methods and
365 **Supp. Data 9**). We found that using the eight approaches, 291 decisions from the differential
366 expression analysis were the same for the 313 shared protein groups. We went on analyzing
367 the 22 differences in detail (**Supp. Data 10**). First, we found that nine of the 22 protein groups
368 were differentially abundant without imputation. Therefore, for these nine protein groups RSN
369 imputation led to non-significant results, whereas the other models found these nine to be
370 significant. The three least significant protein groups without imputation of missing values
371 showed the exact opposite result: Only RSN imputation made these significantly abundant. For
372 three protein groups with q-values just above the 0.05 threshold the self-supervised models and
373 RF imputation lead to differentially abundance, whereas only Median imputation did not.
374
375 As the effect of imputations depends on the clinical covariates in our analysis, we picked six of
376 the 22 protein groups from above (*F7, CFH, IGHV4-4, CRP, CRISP3, DEFA1B/DEFA3*) for
377 detailed analysis (**Table 1**). On one hand when using RSN imputations four protein groups
378 (*IGLV9-49, TTN, ACTA2/ACTG2, F7*) were significant which no other model found, with partly
379 extremly diverging qvalues. Few relatively non-prevalent protein groups, i.e. here below 70%
380 prevalence (250/358 and less) of quantified samples, showed a strong difference in differential

381 analysis testing between e.g. the VAE and the RSN imputation. The relatively non-prevalent
382 (222/358) protein group F5H8B0/P08709/P08709-2 associated with the gene *F7* was clearly not
383 significant using the VAE, but passed the statistical thresholds when using RSN imputation. On
384 the other hand, a relatively rarely quantified protein A0A0D9SG88 (gene *CFH*) was significant
385 using VAE imputation, but not RSN (**Supp. Data 10**). Visualization using swarm plots indicated
386 that the RSN imputation for these two protein groups were shifted downwards of the original
387 center (**Fig. 4d-e**). In combination with a slight imbalance of targets in the 142 imputed samples,
388 this led to the different results. Similarly, for two newly included protein groups P06702 (gene
389 *S100A9*) and Q8WUD1/Q8WUD1-2 (gene *RAB2B*) imputation by a fixed median value did not
390 lead to significant results (**Fig. 4f-g, Supp. Data 10**). It could, therefore, be good scientific
391 practice to check the effect of imputation on differential analysis before a single protein group is
392 selected for further analysis.
393
394 Finally, expanding the analysis to the 64 additional protein groups included here, we found 16 of
395 them to be significantly differentially abundant without any imputation and 36 using VAE
396 imputations (DAE: 35, RF: 37, CF: 35, KNN: 35, Median: 15, see **Supp. Data 9**). Therefore,
397 apart from the difference from the shared protein groups, which yielded 12 more significant hits,
398 imputation using e.g. the VAE allowed us to identify 36 additional protein groups that were not
399 considered for statistical analysis in the original study.
400

Protein Group	F5H8B0 P08709 P08709-2	A0A0D9SG88	A0A075B6R2	P02741	I3L0A1 J3KPA1 P54108 P54108-2 P54108-3	P59665 P59666
Gene	F7	CFH	IGHV4-4	CRP	CRISP3	DEFA1B DEFA3
Q-Value None	0.954	0.116	0.000	0.030	0.018	0.365
Q-Value DAE	0.916	0.032	0.000	0.039	0.005	0.035
Q-Value VAE	0.554	0.025	0.000	0.035	0.003	0.054
Q-Value RF	0.918	0.089	0.000	0.028	0.001	0.087

Q-Value CF	0.895	0.051	0.000	0.037	0.002	0.049
Q-Value KNN	0.884	0.058	0.000	0.028	0.000	0.058
Q-Value Median	0.918	0.112	0.000	0.030	0.051	0.371
Q-Value RSN	0.030	0.943	0.102	0.057	0.746	0.010

Table 1. Examples of diverging decisions between imputation methods. Based on comparison of multiple testing corrected decisions using previously included protein groups. All see **Supp. Data 7,8**.

Robustness of differently abundant protein group identification

Next we decided to analyze in more detail which protein groups were differentially abundant using the different models and how reliable the differential outcome was per model. We repeated the analysis on the ALD data ten times, re-training the models on the same training data split. In this scenario Median, KNN and RSN imputations were guaranteed to yield the same result for all repetitions. Of the 313 originally included protein groups, 25 did not have the same differently abundant outcome for all models on all ten repetitions. Especially for protein groups with few missing values, seeing the same outcome is not unexpected. However, for nine protein groups with two to 29 percent missing values, adding RSN imputations changed the result from being differentially abundant without imputation to being not differentially abundant. DAE, VAE, RF and VAE did not change the result ten times for these nine protein groups. For the newly included protein groups with a higher percentage of missing values 46 of 64 were at least once differentially abundant by using one of the methods. Of these 46, 16 were differentially abundant without imputation, which except by median imputation was not changed using all model based imputation. Additionally, 13 were at least identified seven times as differentially abundant using DAE, VAE, RF or CF imputations (**Supp. Table 2)**. Using CF imputation we found 17 protein groups to be differentially abundant in all ten repetitions. Deterministic KNN imputation in our setup found 19 protein groups to be differential abundant ten times. However, three of these were only found using KNN imputations (**Supp. Data 12**). We conclude that there is some variability in the analysis due to imputations and that for certain imputed protein groups the choice of the imputation method is crucial.

425 **Novel proteins could be biological relevant**

426 We then investigated if these protein groups could be associated with disease using the
427 DISEASES database[37] and found that 39 of the 64 novel proteins had an association entry to
428 fibrosis. Of the 37 new significant protein groups 20 had an association entry to fibrosis, with six
429 having a confidence score greater than two (**Supp. Data 13**). For example, the protein P05362
430 from gene *ICAM1* had the highest disease-association with a score of 3.3. It was found to be
431 significantly dysregulated in the liver data and missing in the plasma data in the original study.
432 Following this reasoning, the second highest scoring protein group was composed of P01033
433 and Q5H9A7 (gene *TIMP1*). This indicated that the novel protein groups using the PIMMS
434 criteria for inclusion could be biologically relevant. In summary, more significant hits were found
435 of which nine had associations with a score greater than two in the DISEASE database
436 indicating the potential relevance of these in ALD.

437 **The additionally added protein groups were predictive of fibrosis**

438 Finally, in the work by Niu et al., the authors trained machine learning models to predict clinical
439 endpoints such as fibrosis from the plasma protein groups. To assess the impact on the
440 machine learning model, a logistic regression, we used the data from the differential analysis
441 above. We replicated the workflow performed by Niu et al., and evaluated the model using the
442 ALD cohort for individuals with histology-based fibrosis staging data available (N=358). Using
443 minimum redundancy, maximum relevance (MRMR) approach we selected the most predictive
444 set of features of each subset of features on the assigned training samples[38]. Using all available
445 protein groups median imputation yielded the best area-under the receiver operating curve
446 (AUROC) of 0.90 compared to 0.87-0.90 for DAE, VAE and CF on the test samples (**Supp. Fig.
447 S8**). For models trained on only the newly included protein groups the DAE and VAE model
448 imputation led to the best AUROC using only one protein group (*TIMP1*) previously mentioned,
449 whereas the model using four median imputed protein groups was worse (VAE: 0.85, DAE:
450 0.84, CF: 0.82, RF: 0.75, Median: 0.70). Therefore, we concluded that the additional protein
451 groups retained with more missing values when using the PIMMS approach compared to the
452 original study's approach were likely relevant protein groups.
453

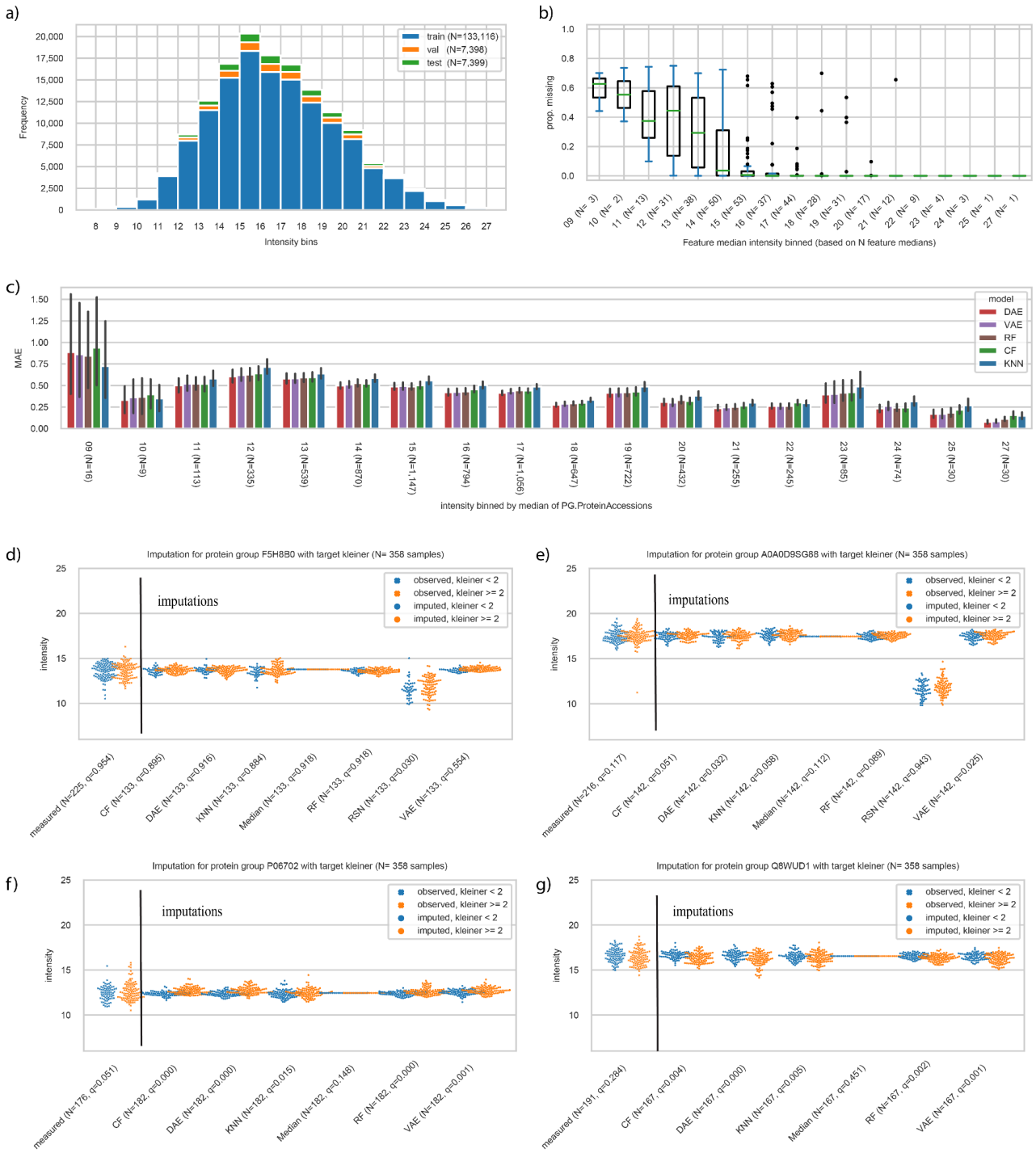

Figure 4: Underlying data distribution and performance in detail on an ALD dataset. a) Stack histogram of all three splits. Bins were created per integer of log2 transformed intensities and number of intensities per split are given in parentheses. **b)** Boxplots of proportion of missing values of protein groups given the integer value of their median intensity. **c)** MAE with 95% confidence interval for simulated intensities in the test split binned by median intensity of their protein group. **d-g)** Comparison of q-values based on no imputation (measured) and adding imputations using the models for four protein groups. True: kleiner score > 1, False: kleiner score < 2.

Discussion

Imputation is an essential step for many analysis types in proteomics, which is often done heuristically. Here we tested three models using a more holistic approach to imputation. We showed that a CF model, a DAE and a VAE reached a similar performance on simulated missing values that represented the entire distribution of the data - including low abundant features. For all comparisons done in this study the performance of the self-supervised models was as good or better than RF or KNN imputations. In comparison to RF it scales better to high dimensional data and outperforms KNN overall, which was especially beneficial when working with peptides instead of protein groups to avoid implicit imputations[14]. Further, we investigated the effect of the imputation method on a concrete analysis, using DIA data from 358 liver patients. Here we found that missing values were imputed by the models towards the lower end of the distribution but less pronounced as when using a heuristic RSN imputation which shifts all replacements towards the limit of detection (LOD) in a sample. We believe that this is due to the lowest abundant features limiting the learned data distributions (**Supp. Fig. S2**) and that some features are not set towards the LOD by the model due to being missing at random. We therefore argue that our holistic model based imputations are more conservative than e.g. the RSN imputations and that the three self-supervised DL models offer a sensible approach to proteomics imputation while scaling well to high feature dimensions. We found that all methods besides RSN were a better choice for the ALD dataset analyzed.

Simulating 20 percent MAR values we saw the ability to recover signal by some models. This suggests that imputation using our or other data driven models can recover lost biological signals which is in line with the observation that significant values are not made insignificant when using data driven models (CF, DAE, VAE, RF, KNN) in comparison to RSN. DAE, VAE and RF performed better than CF and KNN in this application, and median imputation was too heuristic to recover the signal. However, the analysis also revealed that some protein groups will be false positives or negatives as removing some observations can change the outcome. Finally, we offer a workflow to reproduce the comparison done here using any other data provided by the user.

A limitation of the model-based approach is that the models should only be used for imputation if the samples are related. Therefore, the best imputation strategy will be dependent on the experimental setup. We showed that the models can learn to perform imputation on plasma

494 samples from a diverse set of clinical phenotypes ranging from healthy to liver cirrhosis.
495 However, the data-driven models would not perform well when imputing for instance one liver
496 proteome together with ten plasma proteomes. In such a case the data-driven models would not
497 have any other liver proteomes to learn from. If replicas on a small number of samples are the
498 study design, KNN interpolation can be a good remedy or using a set of tests which among
499 others capture missingness[21]. DAE and VAE are not suited to be trained on too few samples,
500 however the exact cutoff will need to be evaluated on a per dataset basis. CF will by the training
501 design not be too dependent on the number of samples if trained sufficiently. In summary, for
502 highly varying features in a complex experimental setting with many differing samples, a holistic
503 model trained with all features and potentially additional related samples can capture
504 dependencies between features - such as the ones implemented in PIMMS.
505
506 In general, the modeling approaches here are restricted to the samples in a particular study and
507 all models are fitted for each new dataset. However, transfer of models between datasets can
508 be envisioned. The potential to fine-tune a model trained on one dataset to a new dataset for a
509 fixed set of features is possible without further efforts for autoencoders. For CF it would need to
510 find the closest training samples in the case where samples are separated strictly into train,
511 validation and test set. However feature embeddings could be transferred and extended easily.
512 Therefore, all models could potentially be envisioned in a clinical setup, where models are re-
513 trained with the latest samples. This could be implemented by the use of similar cohorts, e.g. for
514 the same tissue and similar patients, which is then the basis to build a database of tissue
515 specific models - or by incorporating tissue embeddings as an additional source of information.
516
517 To ensure reproducibility and further extension we offer an evaluation workflow for simulated
518 MAR missing values of the entire data distribution instead of only reporting results on our
519 specific datasets. Everything is available on GitHub, including the workflows, which allows for
520 additional methods to be added to our comparisons. This includes comparison on simulated
521 missing values of the entire data distribution which can be extended to further holistic models.
522 The potential extensibility of the workflow allows for comparison of different ideas on different
523 datasets, including the downstream analysis.
524
525 We evaluated imputation on different levels of proteomics features and found that lower-level
526 data was easier to learn due to being less aggregated[39]. Therefore, it would be great to assess
527 further if machine learning models can be trained on lower-level data as peptides are the most

528 sensible unit and imputation on protein groups level performs one form of implicit imputation at
529 the peptide level[14]. One could assess if imputed features on lower-level data can be
530 reaggregated to protein groups, e.g using ideas from Sticker and coworkers[40]. Additionally, the
531 three self-supervised DL models could also be explored for denoising of samples, especially the
532 generative VAE, or by adding diffusion models as they are trained by adding noise to the data[41].
533
534 Finally, an interesting application will be single cell proteomics. This community in proteomics
535 has not yet developed their own methods to our knowledge, but might not want to fall back to
536 the ones established for discrete count-based single cell RNA data[42,43] for intensity based label-
537 free quantified proteomics data without further testing. In conclusion we suggest that holistic
538 models such as the ones implemented in PIMMS can improve imputation for proteomics and
539 that our evaluation workflow allows further experimentation leading to more robust imputation.

540 Methods

541 **Description of the HeLa proteomics dataset**

542 The HeLa cell lines were repeatedly measured as maintenance (MNT) and quality control (QC)
543 of the mass spectrometers at Novo Nordisk Foundation Center for Protein Research (NNF
544 CPR) and Max Planck Institute of Biochemistry. The samples were run as QC samples during
545 the measurement of cohorts or as MNT samples after instrument cleaning and calibration using
546 different column lengths and liquid chromatography methods. The cells were lysed by different
547 protocols, which are expected to include digestion using trypsin, but on a per sample basis the
548 exact protocol was not annotated[44]. The injection volume ranges from one to seven microliter.
549 Therefore, our dataset contains repeated measures of similar underlying biological samples
550 acquired using DDA label-free quantification and can be used to explore general questions of
551 applicability of self-supervised learning to proteomics data.

552 **Description of raw file processing of HeLa proteomics dataset**

553 We used 564 raw files of quality and maintenance runs of HeLa cell lines from a larger set of
554 7444 quality control and maintenance runs. We processed all of these in a Snakemake[45]
555 workflow as single runs in MaxQuant 1.6.12 [22] yielding single abundances for precursor,
556 aggregated peptide and protein group intensities using LFQ. As FASTA file/Library the
557 UNIPROT human reference proteome database 2018 release, containing 21,007 canonical and

558 72,792 additional sequences, was used for the DDA analysis. Contaminants were controlled
559 using the default contaminants fasta shipped with MaxQuant. From the MaxQuant summary
560 folder we then used the *evidence.txt* for precursor quantifications, *peptides.txt* for aggregated
561 peptides and *proteinGroups.txt* for protein groups. The full dataset and detailed pre-processing
562 steps are explained in a Data Descriptor[46].

563 **Feature selection strategy for quantified runs in MaxQuant**

564 We applied a two-step procedure for feature and sample selection (**Supp. Fig. S1**). We used a
565 cutoff of 25% feature prevalence across samples to be included into the workflow. Samples
566 were then filtered in a second step by their completeness of the selected features. To be
567 included a sample had to have 50% of the selected features. In order to create train, validation
568 and test splits, a dataset was split in the long-data view, where a row consists of a sample
569 name, feature name and its quantification. We divided 90% of the data into training data, 5%
570 into validation and 5% into the test split, adhering to the overall feature frequency. This ensured
571 that the validation and test data were representative of the entire data. The validation cohort
572 was only used for early stopping and the performance on the validation and test data was
573 therefore expected to be similar. On a few hundred sample datasets, the number of sampled
574 quantifications for both validation and test split is quickly in the order of hundred thousand for
575 protein groups and several hundred thousands for peptide-related measurements.

576 **GALA-ALD dataset**

577 The clinical data consisted of a cohort of patients with liver disease[2]. 457 plasma samples were
578 measured in data-independent acquisition (DIA) and processed using Spectronaut v.15.4[47] with
579 the libraries as described in detail in Liu and coworkers[2]. Peptide quantification was extracted
580 from “PEP.Quantity” - representing the stripped peptide sequence. Data for downstream
581 analysis was selected with the same two-step procedure as described for the HeLa data. 3,048
582 aggregated peptides were available in at least 25% of the samples of a total of 4,345
583 aggregated peptides being present at least once. Protein group quantifications were extracted
584 from “PG.Quantity”, dropping filtered-out values. 377 protein groups were available in at least
585 25% of the samples of a total of 506 protein groups being present at least once. We used a
586 fibrosis marker (kleiner[48] score ranging from zero to four, N=358) to compare the effects of
587 different imputation methods on the application. In the original ALD study the features were
588 further selected based on QC samples where a maximum coefficient of variation of 0.4 on the

589 non log transformed qualifications per feature was used as cutoff for inclusion. This step was
590 omitted in the comparison with the original study results in **Fig. 4** as we wanted to have a
591 standardized workflow applicable also to approaches without interspersed QC samples. In
592 numbers this means that we retained 313 protein groups instead of 277 omitting the selection
593 criteria on QC samples. For the differential abundance analysis we had 348 complete clinical
594 samples both with kleiner score and clinical control measurements we used.

595 **Self-supervised DL models**

596 All models used self supervision as their setup, i.e. the data itself is used as a target in a
597 prediction task. CF builds on the idea to combine a sample representation with a feature
598 representation to a target value of interest[49–51]. The simplest implementation is to combine
599 embedding vectors of equal length using their scalar product to the desired outcome, here the
600 log intensity value assigned by a proteomics software program. The approach is flexible to the
601 total number of samples and features, and uses only the non-missing features the model is
602 trained on. The loss function is the mean squared error.
603 A DAE is at inference time a plain autoencoder. During training its input values were partly
604 masked and needed to be reconstructed. For each mini-batch the error was used to update the
605 model so that the model learned better to reconstruct the data[52,53]. The loss was the mean
606 squared error:
607

608 $$L_{reconstruction} = \sum_i^{N_B} \quad \sum_f^{F_i} \quad \left(I_{f,i}^{pred} - I_{f,i}^{obs}\right)^2$$

609

610 where N_B is the number of samples in a batch B, F_i is the number of features not missing in a
611 sample i and $I_{f,i}$ is the predicted and observed label-free quantification intensity value I of
612 feature f in sample i. Missing features in a sample, which were not missing due to the training
613 procedure of masking intensity values, were not used to calculate the loss. This training
614 procedure is also known as contrastive learning and believed to be less stable for training. VAE
615 introduces a different objective and models the latent space explicitly, here and as most often
616 done as a standard normal distribution[54,55]. The latent space of a VAE has two components that
617 are used for the first part of the loss function, the regularization loss:
618

619 $$L_{regularization} = \sum_i^{I_B} \quad \sum_l^{L} \quad max\left(0, 0.5 * \left\{\mu_{l,i}^z + e^{\upsilon_{l,i}^z} - 1 - \upsilon_{l,i}^z\right\}\right)$$

where $\mu_{l,i}^{z}$ is the mean and $v_{l,i}^{z}$ the log variance of dimension l and sample i of the isotropic multivariate Gaussian with L dimensions of the encoder output, i.e. the latent representation z. The reconstruction loss was based assuming a normal distribution for the decoder as output[55,56], leading to

$$L_{reconstruction} = \sum_{i}^{N_B} \sum_{f}^{F_i} 0.5\left\{ln(2\pi) + \left[\left(I_{f,i}^{obs} - \mu_{f,i}^{I}\right)^2\right] + v_{f,i}^{I}\right\}$$

where N_B, F_i and $I_{f,i}^{obs}$ are as before and $\mu_{f,i}^{I}$ and log variance of $v_{f,i}^{I}$ are the parameters of the isotropic multivariate Gaussian distribution of the decoder outputs, i.e. of the modeled feature distribution. Training of the VAE[57] was augmented by masking input values as in the denoising autoencoder (dropout), although this is not strictly necessary due to the stochastic nature of the latent space. For inference, missing values are predicted using both the mean of the encoder and decoder output. The models were implemented using a variety of software including numpy (v.1.20)[58], pandas (v.1.4.)[59,60], pytorch (v.1.10)[61] and fastai (v.2.5)[50].

Other imputation approaches

We used other methods which were available either in scikit-learn or NAGuideR[13]. The latter uses R packages and groups different methods in a larger function. We used KNN interpolation of replicates[24] based on the HeLa cell line measurements being repeated over time. The only parameter to set was how many neighboring samples should be used as replicates. We used three replicates as this was found to be the best setting by Poulous and coauthors[24], which is also the most widely encountered replication number used in the field and set as default of both packages we used. We also used a simple median calculation for each feature across samples. This requires estimating one parameter per feature. For features that did not vary a lot, this strategy should yield robust estimates for missing values. We used a random forest implementation using missForest[35]. The implementation works well for datasets on the protein groups level, but fails for datasets on the peptide and precursor level as these are roughly ten times higher dimensional. We also included methods which assume MNAR missing values, such as the random shifted normal (RSN) distribution for imputation or QRLIC[20]. RSN has as parameters a global mean shift and scaling factor for standard deviation, as well as a mean and standard deviation for each unit of interest, i.e. all quantified features of a sample or for a

651 feature all quantification of that feature across samples. Note, that RSN and other MNAR
652 focused methods assume that measurements are not present as they are below the limit of
653 detection (LOD). However, in our setup of selecting measurements by the entire data
654 distribution of features into the validation and test data splits will lead to many being not from the
655 lower range of intensity values (**Supp. Fig. S9**) and therefore won't yield good results in MAR
656 focused comparison. All methods are available through our workflow and at least one method
657 per included R package is tested via a GitHub action. Some methods and packages did
658 however not work being called as in NAGuideR for some of our data.

659 **Hyperparameter search using simulated missing values**

660 In order to find good configurations for the self-supervised models a grid search was performed
661 on three data levels on the development dataset. We sampled simulated missing values from
662 the dataset, i.e. 5% for validation and 5% for testing. The training procedure and architecture of
663 models was refined using the validation data. The performance of the best performing models
664 on the validation data were then reported using the test data. We found that test performance
665 metrics matched validation metrics up to the second decimal and that many model
666 configurations yielded similar results. Performance was compared between the three self-
667 supervised models to improve performance during model development. In this work all results
668 were reported based on the simulated missing values in the test data split. Different latent
669 representation dimension, namely 10, 25, 50, 75 and 100 dimensions were connected to a
670 varying dimension and composition of hidden layers with a leaky rectified linear activation:
671 (256), (512), (1024), (2056), (128, 64), (256, 128), (512, 256), (512, 512), (512, 256, 128),
672 (1024, 512, 256), (128, 128, 128), (256, 256, 128, 128) - for the encoder and inverted for the
673 decoder. The total number of parameters using these combinations ranged from a couple of
674 ten-thousands in the case of the CF models to tens of millions for the autoencoder
675 architectures. We picked the smallest model in terms of parameters of the top 3 performing
676 ones as their performance was nearly equal on the validation data split. Besides the best
677 models we reported results using other plots. The intensities in a split were binned by the
678 median of the feature, e.g. protein group, they originated from based on the training data split.
679 The MAE per bin was then reported (**Fig. 2e,f**). This is accompanied by a plot showing the
680 proportion of missing values of a feature based on its median value over samples (**Fig. 2c,d**).
681 The correlation plots were based on Pearson correlation of predicted intensities and their
682 original values. The Pearson correlation was calculated for a feature across all predictions of all

683 samples, denoted “per feature correlation”, or for all predictions within one sample, denoted “per
684 sample correlation” (**Supp. Fig. S6**).

685 **Evaluation, imputation and differential expression in GALA-ALD dataset**

686 We used the same splitting approach of the data as for the development dataset for evaluation.
687 We evaluated using a dimension of ten for CF’s sample and feature embeddings, and the DAE
688 and VAE latent spaces. The autoencoders were composed of one hidden layer with 64 neurons
689 both for the encoder and decoder, leading to a total number of parameters between 9,174 and
690 74,462 for the three models. The RSN imputation for the missing values in the original ALD
691 study was done on a per sample basis, i.e. mean and standard deviation for each sample.
692 Using the two-step procedure with defaults as in the original study, this yielded 313 protein
693 groups for comparison (see ALD data description). Using a filtering of 25% for feature
694 prevalence prior to imputation with the VAE (**Supp. Fig. S1**) we increased the share of missing
695 values to 14% for the selected 377 protein groups in comparison to roughly 5% for the 313
696 features using the selection approach as in the original study[2]. The differential analysis was
697 done using an analysis of covariance (ANCOVA) procedure using statsmodels (v.0.12) and
698 pingouin (v.0.5)[62,63]. We used a linear regression with the original kleiner score[48] as the
699 stratification variable of interest for the patient’s cirrhosis disease stage to predict protein
700 quantifications, controlling for covariates. Therefore, effects for each protein group were based
701 on an ANCOVA controlling for age, BMI, gender, steatosis, and abstinence from alcohol as well
702 as correcting for multiple testing as done in the original study. The multiple comparison
703 corrections (q-values) were based on 313 protein groups in the original data imputed using
704 RSN, and on 377 protein groups retained here. Correction for multiple testing correction was
705 done using Benjamini-Hochberg’s correction[64]. The q-values of each DA were then compared
706 for the overlapping 313 protein groups (**Supp. Data 9-11**).

707 **Machine learning in GALA-ALD dataset**

708 In order to assess the predictive performance of newly retained features, we evaluated a logistic
709 regression using different feature sets for the binary target of a fibrosis score greater than one
710 (F2 endpoint in the original study, False: kleiner < 2, True, kleiner ≥ 2). The feature sets were:
711 First, the features retained using the selection approach with settings as in the ALD study;
712 second, all features available when using PIMMS selection approach; third, and the difference
713 between both feature sets termed “new feat”. We used maximum relevance, minimum

redundancy using the F-test based implementation, in detail the F-test correlation quotient (FCQ)[38,65] to select a set of features to be used in the logistic regression. Using cross validation we selected the best set of up to 15 features for each of the three sub datasets. Then, the model was retrained on a final 80-20 percent training-testing data split of samples for each subdataset. Areas under the curve (AUC) for the receiver operation (ROC) and precision recall (PRC) curves were compared between these three sub datasets. The shown graphs and reported metrics were calculated on the test split[62,66,67].

Acknowledgements

We would like to acknowledge Rebeca Quinones, Mario Oroshi and John Damm Sørensen for their huge effort to retrieve all quality and maintenance HeLa raw files for the creation of our development dataset. Furthermore, we would like to thank Jeppe Madsen and Martin Rykær for their time and expertise. Lastly, we would like to thank the proteomics groups that have kindly shared their maintenance and QC HeLa files, both at Novo Nordisk Foundation Center for Protein Research and at the Proteomics and Signal Transduction department of the MaxPlanck institute of biochemistry. H.W. was supported by the Novo Nordisk Foundation (grant NNF19SA0035440). H.W., A.B.N., L.N., M.L-P., M.M, L.J.J. and S.R. were supported by the Novo Nordisk Foundation (grant NNF14CC0001, NNF21SA0072102).

Author contributions

S.R. and A.B.N. initiated the study and guided the analysis. M.M. and A.B.N. led the efforts in collecting the data. H.W. assembled the data, performed the analyses and wrote the software choosing the modeling approaches and refining the idea. M.L.P, L.N. and L.J.J. provided guidance and input for the analysis. H.W. and S.R. wrote the manuscript with contributions from all coauthors. All authors read and approved the final version of the manuscript.

Data availability

The mass spectrometry proteomics data have been deposited to the ProteomeXchange Consortium via the PRIDE[68] partner repository with the dataset identifier PXD042233. A

740 manuscript[44] describing these data has been submitted to Scientific Data and is available as a
741 pre-print at https://doi.org/10.21203/rs.3.rs-3083547/v1.
742

743 Currently the data can be accessed by login for review: https://www.ebi.ac.uk/pride/login

744 - Username: reviewer_pxd042233@ebi.ac.u
745 - Password: DQ7iIJ9N

746 The clinical data is not freely available, but can be requested as specified by Niu et. al.[2] : "The
747 full proteomics datasets and histologic scoring generated and/or analyzed (…) are available (...)
748 upon request, to Odense Patient Data Exploratory Network (open@rsyd.dk) with reference to
749 project ID OP_040. Permission to access and analyze data can be obtained following approval
750 from the Danish Data Protection Agency and the ethics committee for the Region of Southern
751 Denmark."

752 Code availability

753 The PIMMS package and all analysis scripts are available at
754 https://github.com/RasmussenLab/pimms.

755 Competing interest

756 The authors declare no competing interests.

757 References

758 1. Aebersold, R. & Mann, M. Mass-spectrometric exploration of proteome structure and
759 function. *Nature* vol. 537 347–355 Preprint at https://doi.org/10.1038/nature19949 (2016).
760 2. Niu, L. *et al.* Noninvasive proteomic biomarkers for alcohol-related liver disease. *Nat. Med.*
761 **28**, 1277–1287 (2022).
762 3. Francavilla, C. *et al.* Phosphoproteomics of Primary Cells Reveals Druggable Kinase
763 Signatures in Ovarian Cancer. *Cell Rep.* **18**, 3242–3256 (2017).

764 4. Bader, J. M. *et al.* Proteome profiling in cerebrospinal fluid reveals novel biomarkers of
765 Alzheimer's disease. *Mol. Syst. Biol.* **16**, e9356 (2020).
766 5. Schoof, E. M. *et al.* Quantitative single-cell proteomics as a tool to characterize cellular
767 hierarchies. *Nat. Commun.* **12**, 3341 (2021).
768 6. Brunner, A.-D. *et al.* Ultra-high sensitivity mass spectrometry quantifies single-cell
769 proteome changes upon perturbation. *Mol. Syst. Biol.* **18**, e10798 (2022).
770 7. Mund, A. *et al.* Deep Visual Proteomics defines single-cell identity and heterogeneity. *Nat.*
771 *Biotechnol.* **40**, 1231–1240 (2022).
772 8. Meier, F. *et al.* Parallel Accumulation-Serial Fragmentation (PASEF): Multiplying
773 Sequencing Speed and Sensitivity by Synchronized Scans in a Trapped Ion Mobility
774 Device. *J. Proteome Res.* **14**, 5378–5387 (2015).
775 9. Meier, F., Geyer, P. E., Virreira Winter, S., Cox, J. & Mann, M. BoxCar acquisition method
776 enables single-shot proteomics at a depth of 10,000 proteins in 100 minutes. *Nat. Methods*
777 **15**, 440–448 (2018).
778 10. Meier, F., Park, M. A. & Mann, M. Trapped Ion Mobility Spectrometry and Parallel
779 Accumulation-Serial Fragmentation in Proteomics. *Mol. Cell. Proteomics* **20**, 100138
780 (2021).
781 11. Demichev, V., Messner, C. B., Vernardis, S. I., Lilley, K. S. & Ralser, M. DIA-NN: neural
782 networks and interference correction enable deep proteome coverage in high throughput.
783 *Nat. Methods* **17**, 41–44 (2020).
784 12. Webb-Robertson, B.-J. M. *et al.* Review, evaluation, and discussion of the challenges of
785 missing value imputation for mass spectrometry-based label-free global proteomics. *J.*
786 *Proteome Res.* **14**, 1993–2001 (2015).
787 13. Wang, S. *et al.* NAguideR: performing and prioritizing missing value imputations for
788 consistent bottom-up proteomic analyses. *Nucleic Acids Res.* **48**, e83 (2020).
789 14. Lazar, C., Gatto, L., Ferro, M., Bruley, C. & Burger, T. Accounting for the Multiple Natures

790 of Missing Values in Label-Free Quantitative Proteomics Data Sets to Compare Imputation
791 Strategies. *J. Proteome Res.* **15**, 1116–1125 (2016).
792 15. Berg, P., McConnell, E. W., Hicks, L. M., Popescu, S. C. & Popescu, G. V. Evaluation of
793 linear models and missing value imputation for the analysis of peptide-centric proteomics.
794 *BMC Bioinformatics* **20**, 102 (2019).
795 16. Välikangas, T., Suomi, T. & Elo, L. L. A comprehensive evaluation of popular proteomics
796 software workflows for label-free proteome quantification and imputation. *Brief. Bioinform.*
797 **19**, 1344–1355 (2017).
798 17. Čuklina, J. *et al.* Diagnostics and correction of batch effects in large-scale proteomic
799 studies: a tutorial. *Mol. Syst. Biol.* **17**, e10240 (2021).
800 18. Liu, M. & Dongre, A. Proper imputation of missing values in proteomics datasets for
801 differential expression analysis. *Brief. Bioinform.* **22**, (2021).
802 19. Wieczorek, S. *et al.* DAPAR & ProStaR: software to perform statistical analyses in
803 quantitative discovery proteomics. *Bioinformatics* **33**, 135–136 (2017).
804 20. Lazar, C. imputeLCMD: a collection of methods for left-censored missing data imputation. *R*
805 *package, version* **2**, (2015).
806 21. Schwämmle, V., Hagensen, C. E., Rogowska-Wrzesinska, A. & Jensen, O. N. PolySTest:
807 Robust Statistical Testing of Proteomics Data with Missing Values Improves Detection of
808 Biologically Relevant Features. *Mol. Cell. Proteomics* **19**, 1396–1408 (2020).
809 22. Tyanova, S., Temu, T. & Cox, J. The MaxQuant computational platform for mass
810 spectrometry-based shotgun proteomics. *Nat. Protoc.* **11**, 2301–2319 (2016).
811 23. Bouyssié, D. *et al.* Proline: an efficient and user-friendly software suite for large-scale
812 proteomics. *Bioinformatics* **36**, 3148–3155 (2020).
813 24. Poulos, R. C. *et al.* Strategies to enable large-scale proteomics for reproducible research.
814 *Nat. Commun.* **11**, 3793 (2020).
815 25. Nissen, J. N. *et al.* Improved metagenome binning and assembly using deep variational

816 autoencoders. *Nat. Biotechnol.* **39**, 555–560 (2021).
817 26. Frazer, J. *et al.* Disease variant prediction with deep generative models of evolutionary
818 data. *Nature* 1–5 (2021).
819 27. Buergel, T. *et al.* Metabolomic profiles predict individual multidisease outcomes. *Nat. Med.*
820 (2022) doi:10.1038/s41591-022-01980-3.
821 28. Mann, M., Kumar, C., Zeng, W.-F. & Strauss, M. T. Artificial intelligence for proteomics and
822 biomarker discovery. *Cell Syst* **12**, 759–770 (2021).
823 29. Bouwmeester, R., Gabriels, R., Van Den Bossche, T., Martens, L. & Degroeve, S. The Age
824 of Data-Driven Proteomics: How Machine Learning Enables Novel Workflows. *Proteomics*
825 **20**, e1900351 (2020).
826 30. Wen, B. *et al.* Deep Learning in Proteomics. *Proteomics* **20**, e1900335 (2020).
827 31. Bouwmeester, R., Gabriels, R., Hulstaert, N., Martens, L. & Degroeve, S. DeepLC can
828 predict retention times for peptides that carry as-yet unseen modifications. *Nat. Methods*
829 **18**, 1363–1369 (2021).
830 32. Wilhelm, M. *et al.* Deep learning boosts sensitivity of mass spectrometry-based
831 immunopeptidomics. *Nat. Commun.* **12**, 3346 (2021).
832 33. Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, Gilbert Chu. *impute*.
833 (Bioconductor, 2017). doi:10.18129/B9.BIOC.IMPUTE.
834 34. Troyanskaya, O. *et al.* Missing value estimation methods for DNA microarrays.
835 *Bioinformatics* **17**, 520–525 (2001).
836 35. Stekhoven, D. J. & Bühlmann, P. MissForest--non-parametric missing value imputation for
837 mixed-type data. *Bioinformatics* **28**, 112–118 (2012).
838 36. Oba, S. *et al.* A Bayesian missing value estimation method for gene expression profile data.
839 *Bioinformatics* **19**, 2088–2096 (2003).
840 37. Pletscher-Frankild, S., Pallejà, A., Tsafou, K., Binder, J. X. & Jensen, L. J. DISEASES: text
841 mining and data integration of disease-gene associations. *Methods* **74**, 83–89 (2015).

842 38. Zhao, Z., Anand, R. & Wang, M. Maximum Relevance and Minimum Redundancy Feature
843 Selection Methods for a Marketing Machine Learning Platform. *arXiv [stat.ML]* (2019).
844 39. Lazar, C., Gatto, L., Ferro, M., Bruley, C. & Burger, T. Accounting for the Multiple Natures
845 of Missing Values in Label-Free Quantitative Proteomics Data Sets to Compare Imputation
846 Strategies. *J. Proteome Res.* **15**, 1116–1125 (2016).
847 40. Sticker, A., Goeminne, L., Martens, L. & Clement, L. Robust Summarization and Inference
848 in Proteome-wide Label-free Quantification. *Mol. Cell. Proteomics* **19**, 1209–1219 (2020).
849 41. Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. High-Resolution Image
850 Synthesis with Latent Diffusion Models. *arXiv [cs.CV]* (2021).
851 42. van Dijk, D. *et al.* Recovering Gene Interactions from Single-Cell Data Using Data Diffusion.
852 *Cell* **174**, 716–729.e27 (2018).
853 43. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression
854 data analysis. *Genome Biol.* **19**, 15 (2018).
855 44. Webel, H. HeLa quality control sample preparation for MS-based proteomics. *Protocol*
856 *Exchange* (2023) doi:10.21203/rs.3.pex-2155/v1.
857 45. Mölder, F. *et al.* Sustainable data analysis with Snakemake. *F1000Res.* **10**, 33 (2021).
858 46. Webel, H., Perez-Riverol, Y., Nielson, A. B. & Rasmussen, S. Mass spectrometry-based
859 proteomics data from thousands of HeLa control samples. *Research Square* (2023)
860 doi:10.21203/rs.3.rs-3083547/v1.
861 47. Bruderer, R. *et al.* Extending the Limits of Quantitative Proteome Profiling with Data-
862 Independent Acquisition and Application to Acetaminophen-Treated Three-Dimensional
863 Liver Microtissues. *Mol. Cell. Proteomics* **14**, 1400–1410 (2015).
864 48. Kleiner, D. E. *et al.* Design and validation of a histological scoring system for nonalcoholic
865 fatty liver disease. *Hepatology* **41**, 1313–1321 (2005).
866 49. He, X. *et al.* Neural Collaborative Filtering. *arXiv [cs.IR]* (2017)
867 doi:10.1145/3038912.3052569.

50. Howard, J. & Gugger, S. fastai: A Layered API for Deep Learning. *Information* **11**, (2020).
51. Howard, J. & Gugger, S. *Deep Learning for Coders with fastai and PyTorch: AI Applications Without a PhD*. 582 (O'Reilly, 2020).
52. Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P.-A. *Extracting and Composing Robust Features with Denoising Autoencoders*. (2008).
53. Ca, P. V., Edu, L. T., Lajoie, I., Ca, Y. B. & Ca, P.-A. M. *Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion*. vol. 11 3371–3408 http://jmlr.org/papers/v11/vincent10a.html (2010).
54. Kingma, D. P. & Welling, M. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning* **12**, 307–392 (2019).
55. Yu, R. A Tutorial on VAEs: From Bayes' Rule to Lossless Compression. *arXiv [cs.LG]* (2020).
56. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. in *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings* (International Conference on Learning Representations, ICLR, 2014).
57. Im, D. J., Ahn, S., Memisevic, R. & Bengio, Y. Denoising Criterion for Variational Auto-Encoding Framework. *31st AAAI Conference on Artificial Intelligence, AAAI 2017* 2059–2065 (2015).
58. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020).
59. The pandas development team. *pandas-dev/pandas: Pandas*. (2022). doi:10.5281/zenodo.7093122.
60. Mc Kinney, W. Data Structures for Statistical Computing in Python. in *Proceedings of the 9th Python in Science Conference* (eds. van der Walt, S. & Millman, J.) 56–61 (2010).
61. Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32**, (2019).
62. Vallat, R. Pingouin: statistics in Python. *J. Open Source Softw.* **3**, 1026 (2018).

63. Seabold, S. & Perktold, J. Statsmodels: Econometric and statistical modeling with python. in *Proceedings of the 9th Python in Science Conference* vol. 57 10–25080 (Austin, TX, 2010).

64. Benjamini, Y. & Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *J. R. Stat. Soc. Series B Stat. Methodol.* **57**, 289–300 (1995).

65. Mazzanti, S. mrmr-selection. *PyPI* Preprint at https://pypi.org/project/mrmr-selection/0.2.5/ (2022).

66. Seabold, S. & Perktold, J. Statsmodels: Econometric and Statistical Modeling with Python. *9th Python in Science Conference* (2010).

67. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

68. Perez-Riverol, Y. *et al.* The PRIDE database resources in 2022: a hub for mass spectrometry-based proteomics evidences. *Nucleic Acids Res.* **50**, D543–D552 (2022).