Mass spectrometry-based proteomics imputation 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). Some methods only impute assuming the limit of detection (LOD) was not passed and therefore impute missing values with too low or too high intensities, potentially leading to biased results in downstream statistical analysis. Here we test how self supervised deep learning models can impute missing values in the context of LFQ at different levels: precursors, aggregated peptides or protein groups. We demonstrate how collaborative filtering, denoising autoencoders, and variational autoencoders can be used to reconstruct missing values and can make more relevant features available for downstream analysis compared to current approaches. Additionally, we show that deep learning approaches can model data in its entirety for imputation and offer an approach for controlled evaluation of imputation approaches. 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. We identified 49 additional proteins (+23.6%) that are significantly differentially abundant across disease stages compared to traditional methods 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. 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 remains a recurring task for most applications. The noise in data generation 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.

An often-used approach is to impute data at the protein group level 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 normally distributed after log transformation, i.e. entirely determined by their mean and variance. Replacements are then drawn using a normal distribution with a mean shifted towards the lower detection limit with a reduced variance, assuming that proteins are missing due to absence or lower abundance in the sample than the instrument detection limit, which means that the data is left-censored. In this line of thought, several studies focus on determining what works best for different causes of missing values[12,13]. Other studies focus their analysis on post-translational modifications[14], the best combination of software tools, data sets and imputation method[15], normalization and batch effects correction[16] or downstream analysis[17]. 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, linear models[12] or tree-based models[18]. 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. The single cell community in proteomics has not yet developed their own methods to our knowledge, but cannot fall back to established ones for discrete count based single cell RNA data[19,20] for intensity based label-free quantified proteomics data. 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[13,21,22].

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[15,23]. 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[24,25]. 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[26]. 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 and a generic and flexible approach to imputation is not yet established.

We turn to machine learning for imputation 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[27–29]. 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 MS-based proteomics[30–35]. To apply DL methods to imputation of MS-based proteomics data, we considered three types of models that process the inputs 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. The training objectives, complexity, and therefore capabilities of the models are different which led us to evaluate their performance in comparison to each other. 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.

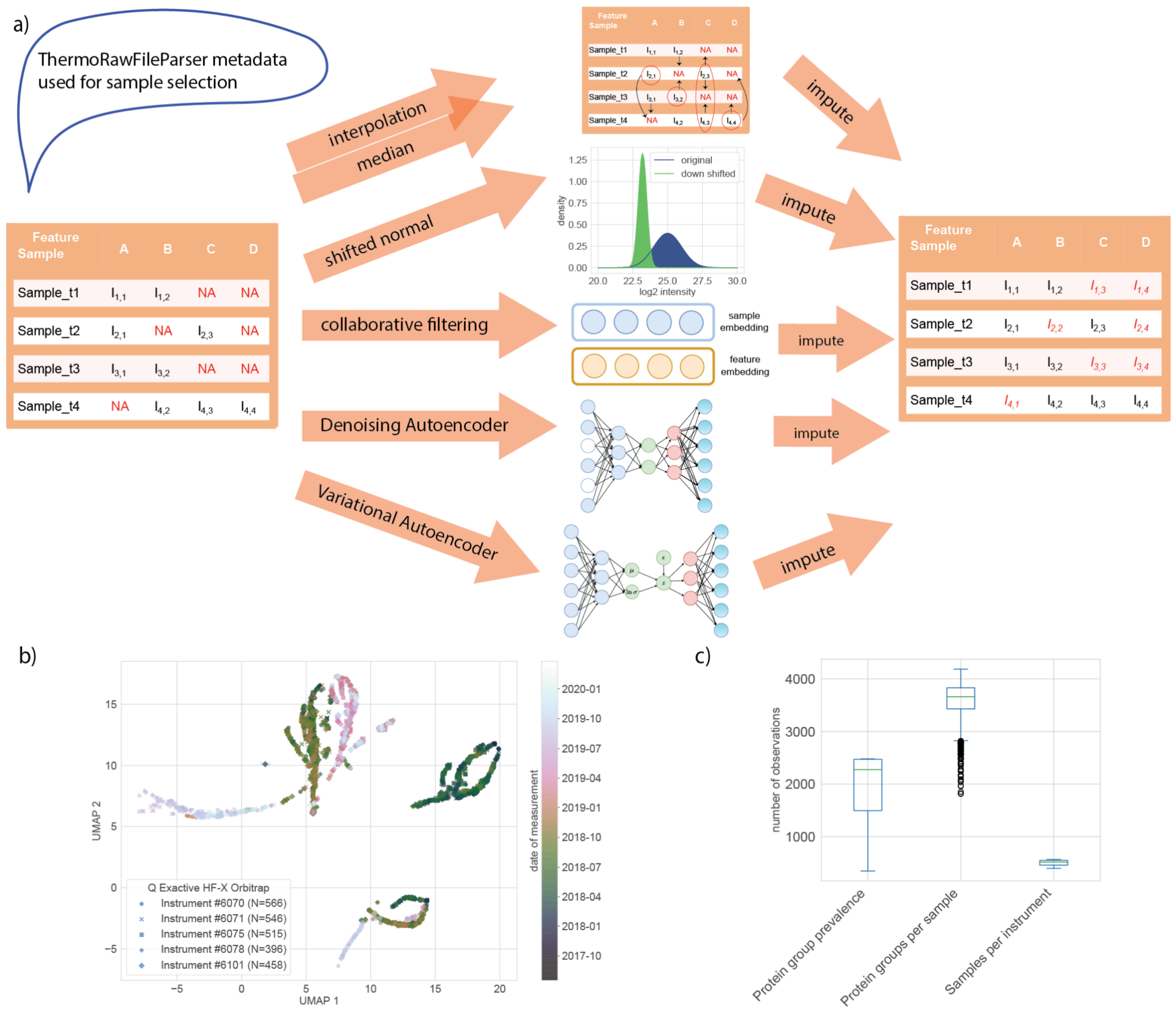
Here we used large (N≈450) and smaller (N≈50) MS-based proteomics data sets of HeLa cell line tryptic lysates acquired on a single machine (Q Executive HF-X Orbitrap) over a period of roughly two years. We applied the DL models described above (CF, DAE, VAE) which create feature representation holistic 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 two heuristic approaches, which could potentially be extended by other methods. Finally, we applied the VAE to a study of ALD patients and found 49 (+23.6%) more significantly differential abundant protein groups 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 two currently used heuristic-based approaches, median imputation per feature across samples and interpolation of missing features by close replicates. While the DL methods and median imputation are able to impute all missing values, interpolation does not replace missing values in case a value is missing in all replicates. Furthermore, interpolation involves repeated measurements of the same samples.

Our development data set initially consisted of 2,481 HeLa cell samples analyzed on five Q Executive HF-X Orbitrap over roughly 3 years, generated during continuous quality control of the mass spectrometers in two different labs. We initially investigated the structure of the data sets using uniform manifold approximation and projection (UMAP)[36] and found that the 2,481 HeLa proteomes on the log-transformed protein groups level clustered according to time (**Fig. 1b**). The median prevalence per protein group, i.e. the median number of samples where a protein group was detected, was 2,275 samples [min: 347, max: 2,481], and samples had a median of 3,663 protein groups [min: 1,815, max: 4,185] (**Fig. 1c).** We ran an imputation comparison based on data from each machine and reported results from the instrument 6070, which had the highest number of samples (566 samples). We used sets of consecutive samples as replicates for the imputation of missing values through interpolation (see Methods), i.e. the samples were ordered by date and three consecutive samples were used as replicates to transfer intensities to missing values. Both heuristic comparison approaches, median imputation per feature across samples and interpolation, did not condition their imputation on the value of other features in a given sample. Validation and test data were drawn according to the entire data distribution from all samples.



Feature Sample	A	B	C	D
Sample_t1	$I_{1,1}$	$I_{1,2}$	NA	NA
Sample_t2	$I_{2,1}$	NA	$I_{2,3}$	NA
Sample_t3	$I_{3,1}$	$I_{3,2}$	NA	NA
Sample_t4	NA	$I_{4,2}$	$I_{4,3}$	$I_{4,4}$

Feature Sample	A	B	C	D
Sample_t1	$I_{1,1}$	$I_{1,2}$	NA	NA
Sample_t2	$I_{2,1}$	NA	$I_{2,3}$	NA
Sample_t3	$I_{3,1}$	$I_{3,2}$	NA	NA
Sample_t4	NA	$I_{4,2}$	$I_{4,3}$	$I_{4,4}$

Feature Sample	A	B	C	D
Sample_t1	$I_{1,1}$	$I_{1,2}$	$I_{1,3}$	$I_{1,4}$
Sample_t2	$I_{2,1}$	$I_{2,2}$	$I_{2,3}$	$I_{2,4}$
Sample_t3	$I_{3,1}$	$I_{3,2}$	$I_{3,3}$	$I_{3,4}$
Sample_t4	$I_{4,1}$	$I_{4,2}$	$I_{4,3}$	$I_{4,4}$

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 two heuristic ones on the development data set: median imputation and replicate interpolation. **b)** UMAP of the five machines with most runs of HeLa data sets from the years of 2017 to 2020 consisting of a total of 2,481 samples, ranging from 396 to 566 samples per instrument. **c)** Summary statistics of 2,481 samples from 5 instruments of HeLa measurement shown in b) for protein groups. Most protein groups associated with a single gene were quantified in thousands of samples. We used a cutoff of 25% feature prevalence across samples to be included into the workflow shown in (a). Samples were then filtered in a second step by their completeness of the selected features (**Fig. S1, Table S1**).

Imputing precursors, aggregated peptides, and protein group data

We applied the imputation methods to a subset of the development data set consisting of 541 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 (see Methods). As the interpolation approach relied on at least one quantified feature in a set of replicates, there could be missing values remaining in cases where all values were missing in a given set. Therefore we restricted the comparison to a subset of imputed simulated missing values by all five approaches: 97,010 of 100,449 for the protein group measurements, 568,397 of 618,711 for the aggregated peptides and 604,377 of 663,028 for the precursors for measurements in the test set for our development dataset of instrument 6070 (**Table S2**). 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 half of the median imputation MAE. This was consistent across the entire distribution of protein group intensities. Interpolation of samples across HeLa cell line measurements had a MAE of 0.76 and the self-supervised models had MAEs of 0.42, 0.43 and 0.42 for CF, DAE and VAE, respectively. Therefore, the median imputation and interpolation models were roughly 1.8-2.4 times worse in comparison to the self-supervised models. Comparing the performance between levels of data aggregation (**Fig. 2a,c,e,f, Supp. Data 1,2**), we found similar patterns as with the protein groups where the self-supervised models reduced the MAE by 42-44% compared to 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 show better performance for lower levels of aggregations[13]. We omitted imputation by RSN as it is not able to model the entire data distribution (see Methods). Additionally, the grid search results showed that the models can be trained without prior normalization and were able to fit the data using many hyperparameter configurations (**Supp. Fig. S2**). Furthermore, we found that the DAE was less stable than the VAE for a wide range of configurations. In summary, this indicated that on an unnormalized, intensity varying data set from a single machine the models were able to capture patterns between detected features to impute values (**Supp. Fig. S3a**) and that the self-supervised models had improved performance for imputing missing values.

Imputation was consistent across feature prevalence

We investigated whether there was a difference in the accuracy of the imputation based on how often a protein group was observed. Here we found that the MAE varied between 0.6 and 0.8 for proteins observed in 25-80% of the samples, whereas for proteins observed in more than 80% of the samples the MAE decreased to below 0.4 (**Fig. 2b**). We observed a similar trend when analyzing a smaller data set of only 50 samples (**Fig. 2d**). Importantly, the MAE for the two heuristic-based approaches, the median and interpolation imputation, was consistently higher as well as following the same pattern. This indicated that some protein groups were harder for all the methods to impute than others, but also that improvement of the CF, DAE, and VAE approaches were not only achieved for some protein groups but consistent across the protein groups. We found similar results for the two other levels of data, aggregated peptides and precursors (**Supp. Fig. S4**). We, therefore, concluded that self-supervised models were able to fit the data holistically for imputation purposes.

Improved 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 (**Fig 2g,h**). The median Pearson correlation between samples was around 0.91 for all self-supervised models and thus by 0.26 higher than the interpolation-based one with 0.65. For features that had less than three observations in the test split, the between-sample correlation was removed. We could not determine correlations for the median imputation as this yields a constant value. The correlation between features within a sample was higher in general, with a median correlation of around 0.97 for all three self-supervised models compared to 0.94 and 0.95 for median imputation and interpolation, respectively. 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.

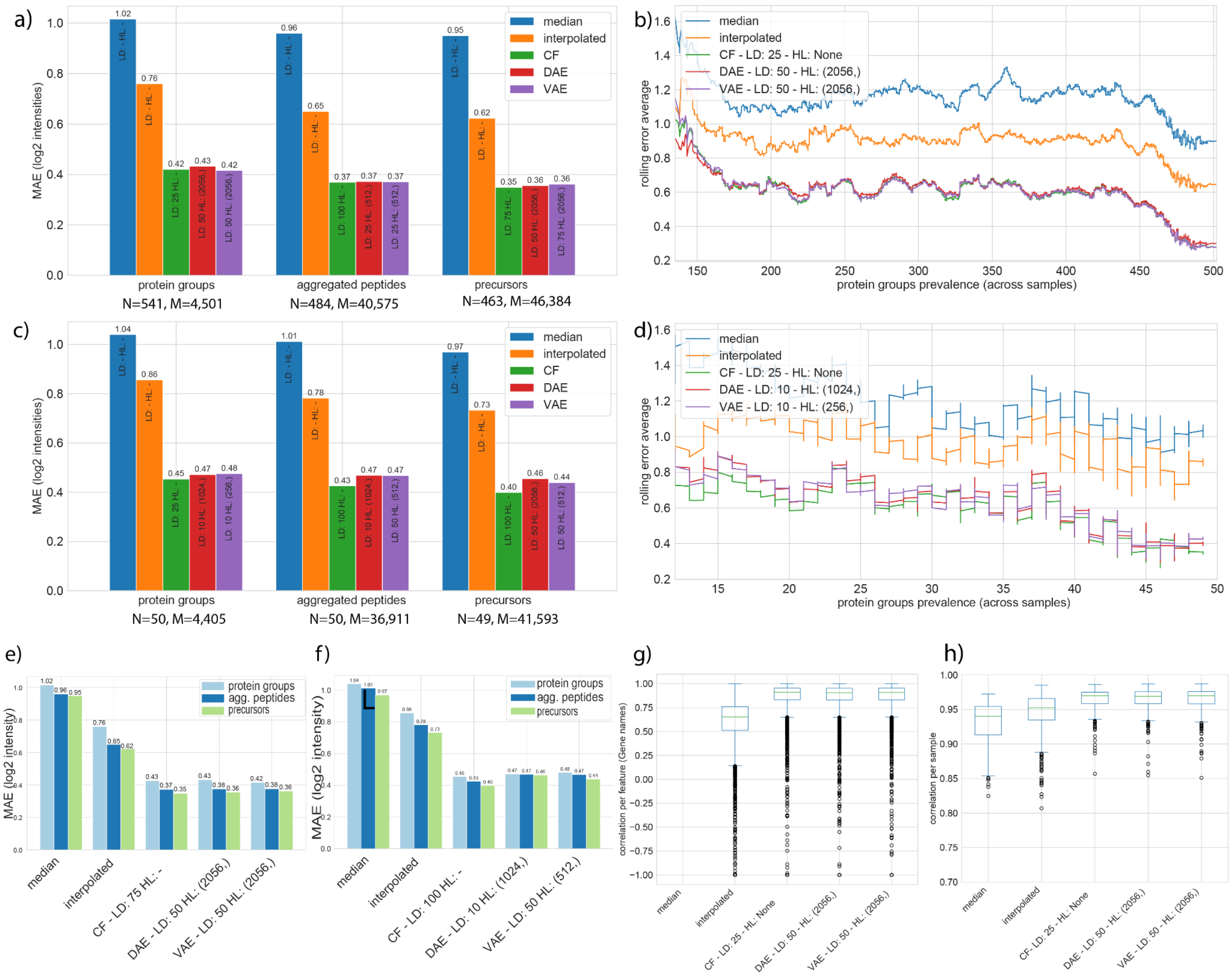


Figure 2: Different levels of MS-based proteomics data can be imputed using self-supervised DL models. a) Performance of imputation methods at the level of protein groups, aggregated peptides, and precursors for MaxQuant outputs. Mean averaged error is shown on the y-axis. Blue: median, yellow: interpolation, green: CF, red: DAE, purple: VAE. The DL methods perform better in comparison to the heuristic-based models. Performance is similar for all three models on each data level and less aggregated data, i.e precursors and aggregated peptides perform better compared to protein groups. **b)** Rolling average error by training data feature frequency for protein groups data (4,491 protein groups, rolling average with the window size of 89). **c)** As (a) but showing a decrease in performance for a subset of the data with only 50 samples. The CF adapts better to smaller sample sizes. **d)** As (b) for protein groups of the smaller data set showing that even few training samples yield good results (3,672 protein groups, rolling average with the window size of 73). **e)** Performance of best average model across data levels, showing nearly identical performance to best models for each data level. Light blue: protein groups, blue: aggregated peptides, green: precursors. **f)** As (e) but showing the smaller dataset. **g)** Boxplot of correlation across samples for each protein group. **h)** Boxplot of correlation across protein groups for each sample, showing that heuristic imputation can also obtain good correlations for protein groups within a sample when there is high variation in the median intensities of protein groups across samples.

Performance of PIMMS on simulated missing values in real use case

To assess the impact of imputation on a large real-world DIA dataset, we applied PIMMS to 455 blood plasma proteomics samples from a cohort of ALD patients and healthy controls[2]. After imputation we again compared how well PIMMS imputed simulated missing values in the ALD data and found that median imputation was only 17% (0.48 vs. 0.41 MAE) worse compared to the best VAE (**Fig. 3a**). This can be explained by the relatively stable measurements of the dataset (**Fig. S3b**) and indicates that most protein groups are stable across patients as the simulated missing values are from the entire distribution of the data. As drawing random replacements from a shifted normal distribution was used in the original study, we add this to our comparison here. However, both the RSN and the interpolation are not really informative for the comparison on the simulated data as basic assumptions are not met. The simulated values are representative of the entire data, and therefore not all low abundant. Similarly, the data does not contain replicates of the same sample, so the interpolation strategy is not entirely sound. In contrast to common practice, the RSN performance shown is on a per feature basis, i.e. using the mean and standard deviation of each feature across samples. If we used the common per sample basis, i.e. using the mean and standard deviation of all features in a sample, the comparison was 4 to 5 times worse on the simulated data. Similarly, the correlation within samples and between samples was best for the self-supervised models. The correlation across samples for each protein group (**Fig. 3c, Supp. Data 3**) was lower for the ALD data in comparison to the heterogeneous measurements of HeLa cell line data for the self-supervised models (median; CF: 0.50, DAE: 0.60, VAE: 0.59). The correlation within samples (**Fig. 3b, Supp. Data 4**) was overall best for the self-supervised models with median correlations of 0.98 (median: NA, CF: 0.98, DAE: 0.98, VAE: 0.98). This, thus, matched the overall results from the HeLa data analysis.

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 the individuals. The fibrosis status was based on liver biopsy, and individuals with all scores ranging from zero to four could be included. By using PIMMS-VAE for imputation of missing data 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]. Both methods replaced missing values with a distribution shifted towards the lower abundance region. However, the maximum intensity for missing values was higher for the VAE with a value of 21 compared to the RSN with a value of 15 for protein groups imputed by both approaches (**Supp. Fig. S5c**). Translating the overall shift in distribution by the VAE to the RSN idea, the intensity distribution was shifted by one standard deviation and the variance was shrinked by 0.7 in comparison to 1.6 standard deviations and 0.3 shrinkage in the ALD-RSN setup. This difference underlies a fundamental difference in the approaches. Whereas the RSN always assumes missing values due to low abundance, the VAE assigns some missing values a higher intensity if other protein groups in the same sample suggest that the missing value occurred rather due to a missed detection than low abundance. When performing differential analysis we found that 208 of the 313 protein groups were significantly differentially expressed using the RSN approach, and 257 of 377 in the PIMMS-VAE setup, this means an increase by 23.6% (see Methods and **Supp. Data 5-8**). We found that using the two imputation approaches, 295 decisions from the differential expression analysis were the same for the 313 shared protein groups. When using the VAE for imputation three protein groups that were significant using RSN were not significant, whereas 15 protein groups were significant using the VAE imputation, but not using RSN (**Fig. 3d**), partly with extremely diverging q-values (**Table 1**). Few relatively non-prevalent protein groups, i.e. here below 70% prevalence (320/455) of quantified samples, showed a strong difference in differential analysis testing between the VAE and the RSN imputation. For instance, the protein group associated with the gene *F7* was clearly not significant using the VAE, but passed the statistical thresholds when using RSN imputation. On the other hand, a relatively rarely quantified protein A0A0D9SG88 (gene *CFH*) was significant using VAE imputation, but not RSN (**Fig. 3d, Supp. Data 6**). It could, therefore, be good scientific practice to check the effect of imputation on differential analysis before a single protein group is selected for further analysis.

Then, expanding the analysis to the 64 additional protein groups imputed by the VAE, we found 37 of them to be significantly differentially abundant. Therefore, apart from the difference from the shared protein groups, which yielded 12 more significant hits, imputation using the VAE allowed us to identify 37 additional protein groups that were not considered for statistical analysis in the original study. We then investigated if these protein groups could be associated with disease using the DISEASES database[37] and found that 39 of the 64 novel proteins had an association entry to fibrosis. Of the 37 new significant protein groups 20 had an association entry to fibrosis, with six having a confidence score greater than two (**Supp. Data 5, 9**). For example, the protein P05362 from gene *ICAM1* had the highest disease-association with a score of 3.3. It was found to be significantly dysregulated in the liver data and missing in the plasma data in the original study. Following this reasoning, the second highest scoring protein group was composed of P01033 and Q5H9A7 (gene *TIMP1*). This indicated that the novel protein groups found when using the VAE for imputation could be biologically relevant. In summary, more significant hits were found of which nine had associations with a score greater than 2 in the DISEASE database indicating the potential relevance of these in ALD.

Protein group	**Gene**	**Q-Value VAE**	**Q-Value RSN**	**UNIPROT – DISEASES (+/- liver disease listed)**

F5H8B0;
P08709;
P08709-2
	F7	0.942	0.047	Coagulation factor VII (+)

P43121	MCAM	0.119	0.029	Cell surface glycoprotein MUC18 (+)
A0A0G2JRQ6	-	0.000	0.538	Ig-like domain-containing protein (-)

I3L0A1;
J3KPA1;
P54108;
P54108-2;
P54108-3
	CRISP3	0.002	0.522	Cysteine-rich secretory protein 3 (-)

Table 1. Examples of diverging decisions between imputation methods. Based on comparison of multiple testing corrected decisions using shared protein groups of RSN and VAE imputation. All see **Supp. Data 6**.

The additional protein groups were predictive of fibrosis

In the work by Niu et al., the authors trained machine learning models to predict clinical endpoints such as fibrosis from the MS plasma protein groups. To assess the impact on the machine learning model, a logistic regression, we used the data from the differential analysis above. We replicated the workflow performed by Niu et al., and evaluated the model using the ALD cohort for individuals with histology-based fibrosis staging data available (N=358). Using minimum redundancy, maximum relevance (MRMR) approach we selected the most predictive set of features of each subset of features[38]. The AUC-ROC values for models trained on all available features using the VAE model (AUC: 0.894) performed as good or slightly better than the one used in the original ALD study (AUC: 0.893). Of the nine predictive protein groups found using the ALD study approach eight overlapped with the predictive protein groups identified using the PIMMS approach (**Supp. Data 10**).

When performing the analysis using only the 64 additional protein groups identified using the PIMMS approach, we found five additional protein groups as predictive for fibrosis (**Fig. 3e,f**). A model trained on only these five protein groups achieved an AUC of 0.753. Three of the new protein groups were also part of the set of protein groups using all VAE-imputed protein groups which showed that some new features selected for prediction were uncorrelated to the original study features. Therefore, we concluded that the additional protein groups identified when using PIMMS compared to RSN were likely relevant protein groups and had high correlation to fibrosis. Retaining more features can allow users to find candidates other researchers might overlook or which have additional predictive power in comparison to already included ones - without compromising on performance.

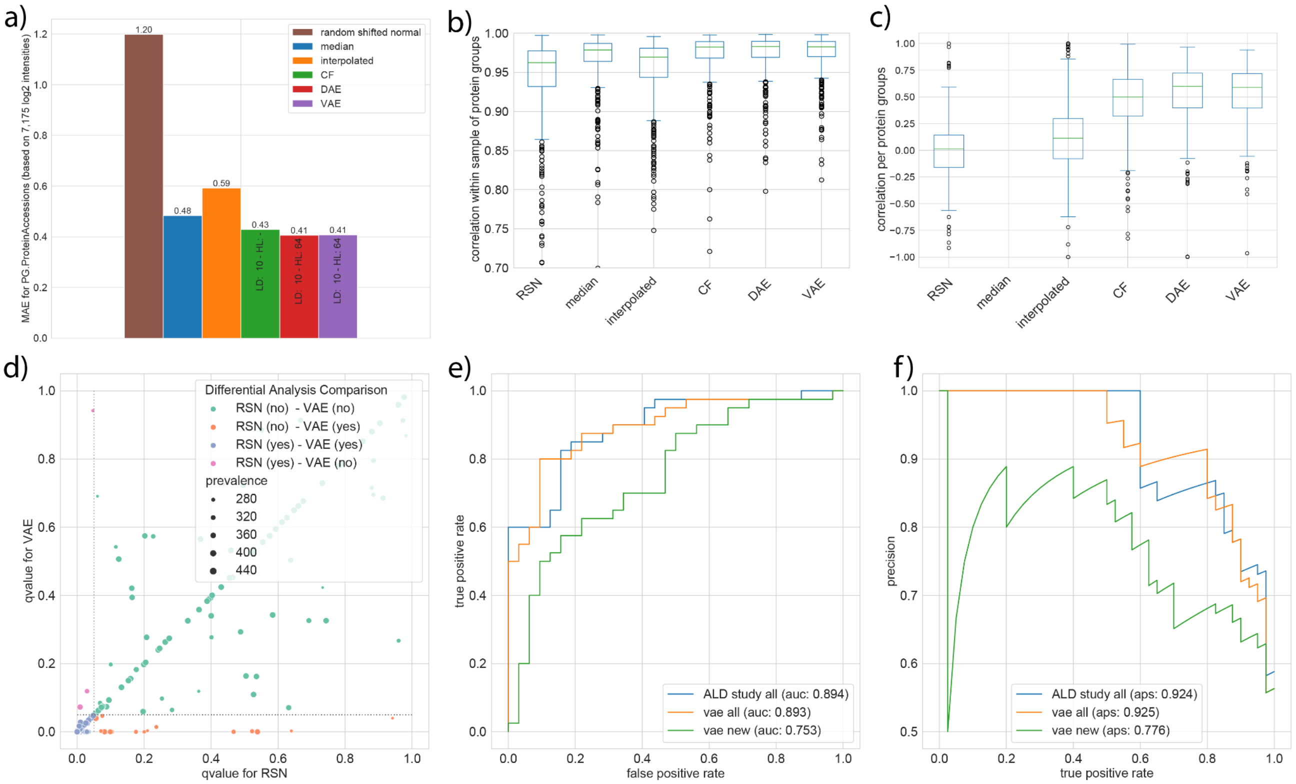


Figure 3: Using PIMMS retains more features with predictive capabilities when applied to a dataset of ALD. a) Performance for protein groups of plasma proteome data on simulated missing values. Interpolation of non-replica was worse than median imputation. Median imputation of simulated test missing values on relatively homogenous data works fairly well. RSN is the worst based on its assumption with regard to missingness. **b)** Within sample Pearson correlation coefficient for imputation of features with simulated missing values. **c)** Between sample Pearson correlation coefficient for imputation of features with simulated missing values. **d)** Comparison of q-values between

shared imputed features when using RSN imputation per sample (x-axis) and the VAE model (y-axis). **e)** ROC-AUC for prediction of fibrosis in ALD patients from plasma proteomics. Blue: original as done by Niu et al., 2022, yellow: using all VAE imputed data, green: using features that were identified additionally when using VAE imputation. **f)** As (e) but showing precision-recall.

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 better than heuristic approaches, which included median, interpolation or shifted normal distribution imputation. Further, we investigated the effect of the imputation method on a concrete analysis, using DIA data from 348 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). We think that this is due to the lowest abundant features limiting the learned data distributions (**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 more sensible approach to proteomics imputation. 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 samples from a diverse set of clinical phenotypes ranging from healthy to liver cirrhosis. However, the models would not perform well when imputing for instance one liver proteome together with ten plasma proteomes. In such a case the model would not have any other liver proteomes to learn from. If replicas on a small number of samples are the study design, interpolation can be a good remedy or using a set of tests which among others capture missingness[23]. For highly varying features in a complex experimental setting with many differing samples, a holistic model trained with all features can capture dependencies between features - such as the ones implemented in PIMMS.

In general, the modeling approaches here are restricted to the samples in a particular study and models are retrained for each new dataset. However, transfer of models between data sets can be envisioned. The potential to fine-tune a model trained on one dataset to a new dataset for a fixed set of features is possible without further efforts for autoencoders. For CF it would need to find the closest training samples in the case where samples are separated strictly into train, validation and test set. However feature embeddings could be transferred and extended easily. Therefore, all models could potentially be envisioned in a clinical setup, where models are re-trained with the latest samples. This could be implemented by the use of similar cohorts, e.g. for the same tissue and similar patients, which is then the basis to build a database of tissue specific models - or by incorporating tissue embeddings as an additional source of information.

To ensure reproducibility and further extension we offer an evaluation workflow for simulated missing values of the entire data distribution instead of only reporting results on our specific data sets. Everything is available on GitHub, including the workflows, which allows for additional methods to be added to our comparisons. This includes comparison on simulated missing values of the entire data distribution which can be extended to further holistic models. The potential extensibility of the workflow allows for comparison of different ideas on different data sets, including the downstream analysis.

We evaluated imputation on different levels of proteomics features and found that lower-level data was easier to learn due to being less aggregated[39]. Therefore, it would be great to assess further if machine learning models can be trained on lower-level data. Alternatively, one could assess if imputed features on lower-level data can be reaggregated to protein groups, e.g using ideas from Sticker and coworkers[40]. Additionally, the three self-supervised DL models could also be employed for denoising of samples, especially the generative VAE, or by adding diffusion models as they are trained by adding noise to the data[41]. Furthermore, one could also use the model to get assignments of values as missing completely at random (MCAR) or missing not at random (MNAR)[42]. Additionally, the self-supervised learning setup can be transferred to different omics.

Finally, an interesting application will be single cell proteomics. This community in proteomics has not yet developed their own methods to our knowledge, but cannot fall back to the ones established for discrete count-based single cell RNA data[19,20] for intensity based label-free quantified proteomics data. In conclusion we suggest that holistic models such as the ones implemented in PIMMS can improve imputation for proteomics and that our evaluation workflow allows further experimentation leading to more robust imputation.

Methods

Description of the HeLa proteomics dataset

The HeLa cell lines were repeatedly measured as maintenance (MNT) and quality control (QC) of the mass spectrometers at Novo Nordisk Foundation Center for Protein Research (NNF CPR) and Max Planck Institute of Biochemistry. The samples were run as QC samples during the measurement of cohorts or as MNT samples after instrument cleaning and calibration using different column lengths and liquid chromatography methods. The cells were lysed by different protocols, which are expected to include digestion using trypsin, but on a per sample basis the exact protocol was not annotated. The injection volume ranges from one to seven microliter. Therefore, a single machine dataset contains repeated measures of similar underlying biological samples and can be used to explore general questions of applicability of self-supervised learning to proteomics data. Most samples were single-shot DDA runs, but there were also a few DIA or fractionated measurements available. In order to stratify the raw files into single machine data sets, we created a workflow to extract the raw file metadata using ThermoRawFileParser[43,44]. This approach gave us five data sets for training with similar machine setup from five Q Exactive HF-X Orbitrap instruments (**Fig. 1b**).

Description of raw file processing of HeLa proteomics dataset

We collected 50,521 raw files of quality and maintenance runs of HeLa cell lines, ranging from a few megabytes to several gigabytes in size. We processed all of these in a Snakemake[43] workflow as single runs in MaxQuant 1.6.12 (ref.[24]) yielding single abundances for precursor, aggregated peptide and protein group intensities using LFQ. As FASTA file/Library the UNIPROT human reference proteome database 2018 release, containing 21,007 canonical and 72,792 additional sequences, was used for the DDA analysis. Contaminants were controlled using the default contaminants fasta shipped with MaxQuant.

Of the 50,521 files we were able to process 11,062 of these one by one with MaxQuant yielding 0 to 54,316 thousand peptides identified. We then selected runs with at least 15,000 identified peptides, which gave us 7,484 runs with a minimum raw file size of 0.618 GB. The resulting 7,484 selected runs could then be grouped by instrument. The hyperparameter plots in **Fig. 2** were based on the data of a single machine, instrument 6070, from this data. For extracting metadata information we processed a total of 50,100 raw files using ThermoFisherRawFileParser. There were a total of 66 unique instruments in the entire dataset based on the instrument name, attribute and serial number, of which 19 had at least 1,000 raw files assigned to them. Note though that the dataset contained fractionated measurements which increased the amount of raw files for some instruments. Among the 7,484 raw files included with a minimum of 15,000 identified peptides there were a total of 37 unique instruments with quantified runs, of which 3 have at least 500 quantified runs. From the MaxQuant summary folder we then used the *evidence.txt* for precursor quantifications, *peptides.txt* for aggregated peptides and *proteinGroups.txt* for protein groups. The full dataset will be described in a companion paper.

Feature selection strategy for quantified runs in MaxQuant

We applied a two-step procedure for feature and sample selection (**Fig. S1**). We used a cutoff of 25% feature prevalence across samples to be included into the workflow. Samples were then filtered in a second step by their completeness of the selected features. To be included a sample had to have 50% of the selected features. To handle the large amount of files we iteratively created a curated dataset. First we counted the available features based on the 7,484 quantified runs. Based on these counts, features with a prevalence of 25% across these runs were selected to be used from single MaxQuant output files. Although we have data from 2010 and onwards, we restricted the training to newer models from the year 2017 to 2020, especially as most runs were available for this period. Then for each file from these years, the selected features were extracted according to the procedure for each data level. For the precursors, i.e. evidence.txt files, we dropped potential contaminant entries, dropped zero intensity entries as they provided no quantification for an identified feature, used the “Intensity” column for the LFQ intensities, used “Sequence” and “Charge” as identifiers, and finally selected the entry with maximum intensity for non unique combinations of “Sequence” and “Charge” as this normally corresponds to the best Andromeda score. It would have been good to include modifications in the selection as this should make the entries unique most of the times, although in rare cases entries are repeated with wide ranging retention times. For aggregated peptides, i.e. peptides.txt files, we used the “Intensity” column for LFQ intensities and used “Sequence” as unique identifiers. Finally, for protein groups, i.e. proteinGroups.txt files, we dropped

"Only_identified_by_site", "Reverse" and "Potential_contaminent" entries, dropped entries without a "Gene name", used "Gene name" as identifier and selected entries with a maximum "Intensity" if one gene is set for more than one proteinGroup. The resulting features per data level are given in **Table S1**.

Data splitting for an experiment

In order to create train, validation and test splits, a data set was split in the long-data view, where a row consists of a sample name, feature name and its quantification. We divided 90% of the data into training data, 5% into validation and 5% into the test split, adhering to the overall feature frequency. This ensured that the validation and test data were representative of the entire data. The validation cohort was only used for early stopping and the performance on the validation and test data was therefore expected to be similar. On a few hundred sample data sets, the number of sampled quantifications for both validation and test split is quickly in the order of hundred thousand for protein groups and several hundred thousands for peptide-related measurements (**Supp. Table S2**).

GALA-ALD dataset

The clinical data consisted of a cohort of patients with liver disease[2]. 457 plasma samples were measured in data-independent acquisition (DIA) and processed using Spectronaut v.15.4[45] with the libraries as described in detail in Liu and coworkers[2]. Peptide quantification was extracted from "PEP.Quantity" - representing the stripped peptide sequence. Data for downstream analysis was selected with the same two-step procedure as described for the HeLa data. 3,048 aggregated peptides were available in at least 25% of the samples of a total of 4,345 aggregated peptides being present at least once. Protein group quantifications were extracted from "PG.Quantity", dropping filtered-out values. Here 11 genes had more than one protein group assigned and were kept in the data. 377 protein groups were available in at least 25% of the samples of a total of 506 protein groups being present at least once. We used a fibrosis marker (kleiner[46] score ranging from zero to four, N=358) to compare the effects of different imputation methods on the application. In the original ALD study the features were further selected based on QC samples where a maximum coefficient of variation of 0.4 on the non log transformed qualifications per feature was used as cutoff for inclusion. This step was omitted in the comparison with the original study results in **Fig. 3c-e** as we wanted to have a standardized workflow applicable also to approaches without interspersed QC samples. In numbers this means that we retained 313 protein groups instead of 277 omitting the selection criteria on QC samples.

Self-supervised DL models

All models used self supervision as their setup, i.e. the data itself is used as a target in a prediction task. CF builds on the idea to combine a sample representation with a feature representation to a target value of interest[47–49]. The simplest implementation is to combine embedding vectors of equal length using their scalar product to the desired outcome, here the log intensity value assigned by a proteomics software program. The approach is flexible to the total number of samples and features, and uses only the non-missing features the model is trained on. The loss function is the mean squared error.

A DAE is at inference time a plain autoencoder. During training its input values were partly masked and needed to be reconstructed. For each mini-batch the error was used to update the model so that the model learned better to reconstruct the data[50,51]. The loss was the mean squared error:

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

where N_B is the number of samples in a batch, F_i is the number of features not missing in a sample and $I_{f,i}$ is the predicted and observed label-free quantification intensity value. Missing features in a sample, which were not missing due to the training procedure of masking intensity values, were not used to calculate the loss. This training procedure is also known as contrastive learning and believed to be less stable for training. VAE introduces a different objective and model the latent space explicitly, here and as most often done as a standard normal distribution[52,53]. The latent space of a VAE has two components that are used for the first part of the loss function, the regularization loss:

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

where $\mu^{z}_{l,i}$ is the mean and $\upsilon^{z}_{l,i}$ the log variance parameters of the isotropic multivariate Gaussian with L dimensions of the encoder output, i.e. the latent representation. The reconstruction loss was based assuming a normal distribution for the decoder as output[53,54], leading to

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

where N_B, F_i and $I^{obs}_{f,i}$ are as before and $\mu^{I}_{f,i}$ and log variance of $\upsilon^{I}_{f,i}$ are the parameters of the isotropic multivariate Gaussian distribution of the decoder outputs, i.e. of the modeled feature distribution. Training of the VAE[55] 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)[56], pandas (v.1.4.)[57,58], pytorch (v.1.10)[59] and fastai (v.2.5)[48].

Heuristic reference approaches

We used three heuristic approaches which did not require training a model. First we used interpolation of replicates[26] 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[26], which is also the most widely encountered replication number used in the field. Using interpolation for imputation, some missing values were not imputed, if for a set of three replicates all values were missing. The set of three replicates for a sample was always the sample measured before and afterwards in time on the same machine. For the first and last sample the closest two samples after or before were taken. Second, we 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. Third, we considered the random shifted normal (RSN) distribution for imputation. We did not use it for simulated missing values as it assumes missingness due to low abundance of features, and is used in software as Perseus[60] or Prostar[21]. It 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 feature all quantification of that feature across samples. Using an RSN distribution allows imputing all missing values.

Hyperparameter search using simulated missing values

In order to find good configurations for the self-supervised models a grid search was performed on three data levels on the development dataset. Sampling simulated missing values from the dataset, i.e. 5% for validation and 5% for testing. The training procedure and architecture of models was refined using the validation data. The performance of the best performing models on the validation data were then reported using the test data. We found that test performance metrics matched validation metrics up to the second decimal. Performance was compared between the three semi-supervised models to improve performance during model development. In the article all results were reported based on the simulated missing values in the test data split. Different latent representation dimension, namely 10, 25, 50, 75 and 100 dimensions were connected to a varying dimension and composition of hidden layers with a leaky rectified linear activation: (256), (512), (1024), (2056), (128, 64), (256, 128), (512, 256), (512, 512), (512, 256, 128), (1024, 512, 256), (128, 128, 128), (256, 256, 128, 128) - for the encoder and inverted for the decoder. The total number of parameters using these combinations ranged from a couple of ten-thousands in the case of the CF models to tens of millions for the autoencoder architectures. Besides the best models we reported results using three other plots. The rolling average plots were created with a window size of the number of features divided by 50 for the development dataset over the mean absolute error per feature. The features were ordered by their prevalence in the development dataset and the MAE was calculated for all intensities for a given feature in the split. The best average setup (**Fig. 2e,f**) was determined by finding the best average for a combination of hyperparameters over three data levels, i.e protein groups, aggregated peptides and precursors. The correlation plots were based on Pearson correlation of predicted intensities and their original values. The Pearson correlation was calculated for a feature across all predictions of all samples, denoted “per feature correlation”, or for all predictions within one sample, denoted “per sample correlation”. Finally, we did not include imputation by randomly drawing replacements from a shifted normal distribution, short RSN imputation, as it would always perform worse by design because the underlying assumption for RSN imputation is that measurements are not present as they are below the limit of detection (LOD). However, the selection of measurements by the entire data distribution of features into the validation and test data splits will lead to many being not from the lower range of intensity values.

Evaluation, imputation and differential expression in GALA-ALD dataset

We used the same splitting approach of the data as for the development dataset for evaluation. We evaluated using a dimension of 10 for CF's sample and feature embeddings, and the DAE and VAE latent spaces. The Autoencoders were composed of one hidden layer with 64 neurons both for the encoder and decoder, leading to a total number of parameters between 9,174 and 74,462 for the three models. The imputation for the missing values in the original ALD study was done on a per sample basis, i.e. mean and standard deviation for each sample. Using the two-step procedure with defaults as in the original study this yielded 313 protein groups for comparison (see ALD data description). Using a filtering of 25% for feature prevalence prior to imputation with the VAE (**Fig. S1**) we increased the share of missing values to 14% for the selected 377 protein groups in comparison to roughly 5% for the 313 features using the selection approach as in the original study[2]. Protein group data matrices were imputed using PIMMS-VAE or RSN prior to differential analysis (DA). The VAE had one hidden layer in the encoder and decoder with 64 latent neurons connecting a ten dimensional hidden layer. The differential analysis was done using an analysis of covariance (ANCOVA) procedure using statsmodels (v.0.12) and pingouin (v.0.5)[61,62]. We used a linear regression with the original kleiner score[46] as the stratification variable of interest for the patient's cirrhosis disease stage to predict protein quantifications, controlling for covariates. Therefore, effects for each protein group were based on an ANCOVA controlling for age, BMI, gender, steatosis, and abstinence from alcohol as well as correcting for multiple testing as done in the original study. The multiple comparison corrections (q-values) were based on 313 protein groups in the original data imputed using RSN, and on 377 protein groups retained using the PIMMS-VAE. Correction for multiple testing correction was done using Benjamini-Hochberg's correction[63] based on a varying number of tests. The q-values of each DA were then compared for the overlapping 313 protein groups (**Supp. Data 7**).

Machine learning in GALA-ALD dataset

In order to assess the predictive performance of newly retained features, we evaluated a logistic regression using different feature sets for the binary target of a fibrosis score greater than one (F2 endpoint in the original study). The feature sets were: the features retained using the selection approach with settings as in the ALD study; all features available when using PIMMS selection approach; and the difference 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)[38,64] 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 data sets. 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 data sets. The shown graphs and reported metrics were calculated on the test split[61,65,66]**.**

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 data set. 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 raw and processed development data will be available at PRIDE[67] upon acceptance of the manuscript. The clinical data is not freely available, but can be requested as specified by Niu et. al.[2] : “The full proteomics datasets and histologic scoring generated and/or analyzed (…) are available (...) upon request, to Odense Patient Data Exploratory Network (open@rsyd.dk) with reference to project ID OP_040. Permission to access and analyze data can be obtained following approval from the Danish Data Protection Agency and the ethics committee for the Region of Southern Denmark."

Code availability

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

Competing interest

The authors declare no competing interests.

References

(1) Aebersold, R.; Mann, M. Mass-Spectrometric Exploration of Proteome Structure and Function. *Nature*. Nature Publishing Group September 14, 2016, pp 347–355. https://doi.org/10.1038/nature19949.

(2) Niu, L.; Thiele, M.; Geyer, P. E.; Rasmussen, D. N.; Webel, H. E.; Santos, A.; Gupta, R.; Meier, F.; Strauss, M.; Kjaergaard, M.; Lindvig, K.; Jacobsen, S.; Rasmussen, S.; Hansen, T.; Krag, A.; Mann, M. Noninvasive Proteomic Biomarkers for Alcohol-Related Liver Disease. *Nat. Med.* **2022**, *28* (6), 1277–1287. https://doi.org/10.1038/s41591-022-01850-y.

(3) Francavilla, C.; Lupia, M.; Tsafou, K.; Villa, A.; Kowalczyk, K.; Rakownikow Jersie-Christensen, R.; Bertalot, G.; Confalonieri, S.; Brunak, S.; Jensen, L. J.; Cavallaro, U.; Olsen, J. V. Phosphoproteomics of Primary Cells Reveals Druggable Kinase Signatures in Ovarian Cancer. *Cell Rep.* **2017**, *18* (13), 3242–3256. https://doi.org/10.1016/j.celrep.2017.03.015.

(4) Bader, J. M.; Geyer, P. E.; Müller, J. B.; Strauss, M. T.; Koch, M.; Leypoldt, F.; Koertvelyessy, P.; Bittner, D.; Schipke, C. G.; Incesoy, E. I.; Peters, O.; Deigendesch, N.; Simons, M.; Jensen, M. K.; Zetterberg, H.; Mann, M. Proteome Profiling in Cerebrospinal Fluid Reveals Novel Biomarkers of Alzheimer's Disease. *Mol. Syst. Biol.* **2020**, *16* (6), e9356. https://doi.org/10.15252/msb.20199356.

(5) Schoof, E. M.; Furtwängler, B.; Üresin, N.; Rapin, N.; Savickas, S.; Gentil, C.; Lechman, E.; Keller, U. A. D.; Dick, J. E.; Porse, B. T. Quantitative Single-Cell Proteomics as a Tool to Characterize Cellular Hierarchies. *Nat. Commun.* **2021**, *12* (1), 3341. https://doi.org/10.1038/s41467-021-23667-y.

(6) Brunner, A.-D.; Thielert, M.; Vasilopoulou, C.; Ammar, C.; Coscia, F.; Mund, A.; Hoerning, O. B.; Bache, N.; Apalategui, A.; Lubeck, M.; Richter, S.; Fischer, D. S.; Raether, O.; Park, M. A.; Meier, F.; Theis, F. J.; Mann, M. Ultra-High Sensitivity Mass Spectrometry Quantifies Single-Cell Proteome Changes upon Perturbation. *Mol. Syst. Biol.* **2022**, *18* (3), e10798. https://doi.org/10.15252/msb.202110798.

(7) Mund, A.; Coscia, F.; Kriston, A.; Hollandi, R.; Kovács, F.; Brunner, A.-D.; Migh, E.;

Schweizer, L.; Santos, A.; Bzorek, M.; Naimy, S.; Rahbek-Gjerdrum, L. M.; Dyring-Andersen, B.; Bulkescher, J.; Lukas, C.; Eckert, M. A.; Lengyel, E.; Gnann, C.; Lundberg, E.; Horvath, P.; Mann, M. Deep Visual Proteomics Defines Single-Cell Identity and Heterogeneity. *Nat. Biotechnol.* **2022**, *40* (8), 1231–1240. https://doi.org/10.1038/s41587-022-01302-5.

(8) Meier, F.; Beck, S.; Grassl, N.; Lubeck, M.; Park, M. A.; Raether, O.; Mann, M. Parallel Accumulation-Serial Fragmentation (PASEF): Multiplying Sequencing Speed and Sensitivity by Synchronized Scans in a Trapped Ion Mobility Device. *J. Proteome Res.* **2015**, *14* (12), 5378–5387. https://doi.org/10.1021/acs.jproteome.5b00932.

(9) Meier, F.; Geyer, P. E.; Virreira Winter, S.; Cox, J.; Mann, M. BoxCar Acquisition Method Enables Single-Shot Proteomics at a Depth of 10,000 Proteins in 100 Minutes. *Nat. Methods* **2018**, *15* (6), 440–448. https://doi.org/10.1038/s41592-018-0003-5.

(10) Meier, F.; Park, M. A.; Mann, M. Trapped Ion Mobility Spectrometry and Parallel Accumulation-Serial Fragmentation in Proteomics. *Mol. Cell. Proteomics* **2021**, *20*, 100138. https://doi.org/10.1016/j.mcpro.2021.100138.

(11) Demichev, V.; Messner, C. B.; Vernardis, S. I.; Lilley, K. S.; Ralser, M. DIA-NN: Neural Networks and Interference Correction Enable Deep Proteome Coverage in High Throughput. *Nat. Methods* **2020**, *17* (1), 41–44. https://doi.org/10.1038/s41592-019-0638-x.

(12) Webb-Robertson, B.-J. M.; Wiberg, H. K.; Matzke, M. M.; Brown, J. N.; Wang, J.; McDermott, J. E.; Smith, R. D.; Rodland, K. D.; Metz, T. O.; Pounds, J. G.; Waters, K. M. Review, Evaluation, and Discussion of the Challenges of Missing Value Imputation for Mass Spectrometry-Based Label-Free Global Proteomics. *J. Proteome Res.* **2015**, *14* (5), 1993–2001. https://doi.org/10.1021/pr501138h.

(13) Lazar, C.; Gatto, L.; Ferro, M.; Bruley, C.; Burger, T. Accounting for the Multiple Natures of Missing Values in Label-Free Quantitative Proteomics Data Sets to Compare Imputation Strategies. *J. Proteome Res.* **2016**, *15* (4), 1116–1125. https://doi.org/10.1021/acs.jproteome.5b00981.

(14) Berg, P.; McConnell, E. W.; Hicks, L. M.; Popescu, S. C.; Popescu, G. V. Evaluation of Linear Models and Missing Value Imputation for the Analysis of Peptide-Centric Proteomics. *BMC Bioinformatics* **2019**, *20* (Suppl 2), 102. https://doi.org/10.1186/s12859-019-2619-6.

(15) Välikangas, T.; Suomi, T.; Elo, L. L. A Comprehensive Evaluation of Popular Proteomics Software Workflows for Label-Free Proteome Quantification and Imputation. *Brief. Bioinform.* **2017**, *19* (6), 1344–1355. https://doi.org/10.1093/bib/bbx054.

(16) Čuklina, J.; Lee, C. H.; Williams, E. G.; Sajic, T.; Collins, B. C.; Rodríguez Martínez, M.; Sharma, V. S.; Wendt, F.; Goetze, S.; Keele, G. R.; Wollscheid, B.; Aebersold, R.; Pedrioli, P. G. A. Diagnostics and Correction of Batch Effects in Large-Scale Proteomic Studies: A Tutorial. *Mol. Syst. Biol.* **2021**, *17* (8), e10240. https://doi.org/10.15252/msb.202110240.

(17) Liu, M.; Dongre, A. Proper Imputation of Missing Values in Proteomics Datasets for Differential Expression Analysis. *Brief. Bioinform.* **2021**, *22* (3). https://doi.org/10.1093/bib/bbaa112.

(18) Wang, S.; Li, W.; Hu, L.; Cheng, J.; Yang, H.; Liu, Y. NAguideR: Performing and Prioritizing Missing Value Imputations for Consistent Bottom-up Proteomic Analyses. *Nucleic Acids Res.* **2020**, *48* (14), e83. https://doi.org/10.1093/nar/gkaa498.

(19) van Dijk, D.; Sharma, R.; Nainys, J.; Yim, K.; Kathail, P.; Carr, A. J.; Burdziak, C.; Moon, K. R.; Chaffer, C. L.; Pattabiraman, D.; Bierie, B.; Mazutis, L.; Wolf, G.; Krishnaswamy, S.; Pe'er, D. Recovering Gene Interactions from Single-Cell Data Using Data Diffusion. *Cell* **2018**, *174* (3), 716–729.e27. https://doi.org/10.1016/j.cell.2018.05.061.

(20) Wolf, F. A.; Angerer, P.; Theis, F. J. SCANPY: Large-Scale Single-Cell Gene Expression Data Analysis. *Genome Biol.* **2018**, *19* (1), 15. https://doi.org/10.1186/s13059-017-1382-0.

(21) Wieczorek, S.; Combes, F.; Lazar, C.; Giai Gianetto, Q.; Gatto, L.; Dorffer, A.; Hesse, A.-M.; Couté, Y.; Ferro, M.; Bruley, C.; Burger, T. DAPAR & ProStaR: Software to Perform

Statistical Analyses in Quantitative Discovery Proteomics. *Bioinformatics* **2017**, *33* (1), 135–136. https://doi.org/10.1093/bioinformatics/btw580.

(22) Lazar, C. imputeLCMD: A Collection of Methods for Left-Censored Missing Data Imputation. *R package, version* **2015**, *2*.

(23) Schwämmle, V.; Hagensen, C. E.; Rogowska-Wrzesinska, A.; Jensen, O. N. PolySTest: Robust Statistical Testing of Proteomics Data with Missing Values Improves Detection of Biologically Relevant Features. *Mol. Cell. Proteomics* **2020**, *19* (8), 1396–1408. https://doi.org/10.1074/mcp.RA119.001777.

(24) Tyanova, S.; Temu, T.; Cox, J. The MaxQuant Computational Platform for Mass Spectrometry-Based Shotgun Proteomics. *Nat. Protoc.* **2016**, *11* (12), 2301–2319. https://doi.org/10.1038/nprot.2016.136.

(25) Bouyssié, D.; Hesse, A.-M.; Mouton-Barbosa, E.; Rompais, M.; Macron, C.; Carapito, C.; Gonzalez de Peredo, A.; Couté, Y.; Dupierris, V.; Burel, A.; Menetrey, J.-P.; Kalaitzakis, A.; Poisat, J.; Romdhani, A.; Burlet-Schiltz, O.; Cianférani, S.; Garin, J.; Bruley, C. Proline: An Efficient and User-Friendly Software Suite for Large-Scale Proteomics. *Bioinformatics* **2020**, *36* (10), 3148–3155. https://doi.org/10.1093/bioinformatics/btaa118.

(26) Poulos, R. C.; Hains, P. G.; Shah, R.; Lucas, N.; Xavier, D.; Manda, S. S.; Anees, A.; Koh, J. M. S.; Mahboob, S.; Wittman, M.; Williams, S. G.; Sykes, E. K.; Hecker, M.; Dausmann, M.; Wouters, M. A.; Ashman, K.; Yang, J.; Wild, P. J.; deFazio, A.; Balleine, R. L.; Tully, B.; Aebersold, R.; Speed, T. P.; Liu, Y.; Reddel, R. R.; Robinson, P. J.; Zhong, Q. Strategies to Enable Large-Scale Proteomics for Reproducible Research. *Nat. Commun.* **2020**, *11* (1), 3793. https://doi.org/10.1038/s41467-020-17641-3.

(27) Nissen, J. N.; Johansen, J.; Allesøe, R. L.; Sønderby, C. K.; Armenteros, J. J. A.; Grønbech, C. H.; Jensen, L. J.; Nielsen, H. B.; Petersen, T. N.; Winther, O.; Rasmussen, S. Improved Metagenome Binning and Assembly Using Deep Variational Autoencoders. *Nat. Biotechnol.* **2021**, *39* (5), 555–560. https://doi.org/10.1038/s41587-020-00777-4.

(28) Frazer, J.; Notin, P.; Dias, M.; Gomez, A.; Min, J. K.; Brock, K.; Gal, Y.; Marks, D. S. Disease Variant Prediction with Deep Generative Models of Evolutionary Data. *Nature* **2021**, 1–5. https://doi.org/10.1038/s41586-021-04043-8.

(29) Buergel, T.; Steinfeldt, J.; Ruyoga, G.; Pietzner, M.; Bizzarri, D.; Vojinovic, D.; Upmeier Zu Belzen, J.; Loock, L.; Kittner, P.; Christmann, L.; Hollmann, N.; Strangalies, H.; Braunger, J. M.; Wild, B.; Chiesa, S. T.; Spranger, J.; Klostermann, F.; van den Akker, E. B.; Trompet, S.; Mooijaart, S. P.; Sattar, N.; Jukema, J. W.; Lavrijssen, B.; Kavousi, M.; Ghanbari, M.; Ikram, M. A.; Slagboom, E.; Kivimaki, M.; Langenberg, C.; Deanfield, J.; Eils, R.; Landmesser, U. Metabolomic Profiles Predict Individual Multidisease Outcomes. *Nat. Med.* **2022**. https://doi.org/10.1038/s41591-022-01980-3.

(30) Mann, M.; Kumar, C.; Zeng, W.-F.; Strauss, M. T. Artificial Intelligence for Proteomics and Biomarker Discovery. *Cell Syst* **2021**, *12* (8), 759–770. https://doi.org/10.1016/j.cels.2021.06.006.

(31) Bouwmeester, R.; Gabriels, R.; Van Den Bossche, T.; Martens, L.; Degroeve, S. The Age of Data-Driven Proteomics: How Machine Learning Enables Novel Workflows. *Proteomics* **2020**, *20* (21-22), e1900351. https://doi.org/10.1002/pmic.201900351.

(32) Wen, B.; Zeng, W.-F.; Liao, Y.; Shi, Z.; Savage, S. R.; Jiang, W.; Zhang, B. Deep Learning in Proteomics. *Proteomics* **2020**, *20* (21-22), e1900335. https://doi.org/10.1002/pmic.201900335.

(33) Bouwmeester, R.; Gabriels, R.; Hulstaert, N.; Martens, L.; Degroeve, S. DeepLC Can Predict Retention Times for Peptides That Carry as-yet Unseen Modifications. *Nat. Methods* **2021**, *18* (11), 1363–1369. https://doi.org/10.1038/s41592-021-01301-5.

(34) Declercq, A.; Bouwmeester, R.; Hirschler, A.; Carapito, C.; Degroeve, S.; Martens, L.; Gabriels, R. MS2Rescore: Data-Driven Rescoring Dramatically Boosts Immunopeptide Identification Rates. *bioRxiv*, 2022, 2021.11.02.466886.

https://doi.org/10.1101/2021.11.02.466886.

(35) Wilhelm, M.; Zolg, D. P.; Graber, M.; Gessulat, S.; Schmidt, T.; Schnatbaum, K.; Schwencke-Westphal, C.; Seifert, P.; de Andrade Krätzig, N.; Zerweck, J.; Knaute, T.; Bräunlein, E.; Samaras, P.; Lautenbacher, L.; Klaeger, S.; Wenschuh, H.; Rad, R.; Delanghe, B.; Huhmer, A.; Carr, S. A.; Clauser, K. R.; Krackhardt, A. M.; Reimer, U.; Kuster, B. Deep Learning Boosts Sensitivity of Mass Spectrometry-Based Immunopeptidomics. *Nat. Commun.* **2021**, *12* (1), 3346. https://doi.org/10.1038/s41467-021-23713-9.

(36) Mcinnes, L.; Healy, J.; Melville, J. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*; 2018.

(37) Pletscher-Frankild, S.; Pallejà, A.; Tsafou, K.; Binder, J. X.; Jensen, L. J. DISEASES: Text Mining and Data Integration of Disease-Gene Associations. *Methods* **2015**, *74*, 83–89. https://doi.org/10.1016/j.ymeth.2014.11.020.

(38) Zhao, Z.; Anand, R.; Wang, M. Maximum Relevance and Minimum Redundancy Feature Selection Methods for a Marketing Machine Learning Platform. *arXiv [stat.ML]*, 2019. http://arxiv.org/abs/1908.05376.

(39) Lazar, C.; Gatto, L.; Ferro, M.; Bruley, C.; Burger, T. Accounting for the Multiple Natures of Missing Values in Label-Free Quantitative Proteomics Data Sets to Compare Imputation Strategies. *J. Proteome Res.* **2016**, *15* (4), 1116–1125. https://doi.org/10.1021/acs.jproteome.5b00981.

(40) Sticker, A.; Goeminne, L.; Martens, L.; Clement, L. Robust Summarization and Inference in Proteome-Wide Label-Free Quantification. *Mol. Cell. Proteomics* **2020**, *19* (7), 1209–1219. https://doi.org/10.1074/mcp.RA119.001624.

(41) Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. *arXiv [cs.CV]*, 2021. http://arxiv.org/abs/2112.10752.

(42) Gianetto, Q. G.; Wieczorek, S.; Couté, Y.; Burger, T. A Peptide-Level Multiple Imputation Strategy Accounting for the Different Natures of Missing Values in Proteomics Data. *bioRxiv* **2020**, 2020.05.29.122770. https://doi.org/10.1101/2020.05.29.122770.

(43) Mölder, F.; Jablonski, K. P.; Letcher, B.; Hall, M. B.; Tomkins-Tinch, C. H.; Sochat, V.; Forster, J.; Lee, S.; Twardziok, S. O.; Kanitz, A.; Wilm, A.; Holtgrewe, M.; Rahmann, S.; Nahnsen, S.; Köster, J. Sustainable Data Analysis with Snakemake. *F1000Res.* **2021**, *10*, 33. https://doi.org/10.12688/f1000research.29032.1.

(44) Hulstaert, N.; Shofstahl, J.; Sachsenberg, T.; Walzer, M.; Barsnes, H.; Martens, L.; Perez-Riverol, Y. ThermoRawFileParser: Modular, Scalable, and Cross-Platform RAW File Conversion. *J. Proteome Res.* **2020**, *19* (1), 537–542. https://doi.org/10.1021/acs.jproteome.9b00328.

(45) Bruderer, R.; Bernhardt, O. M.; Gandhi, T.; Miladinović, S. M.; Cheng, L.-Y.; Messner, S.; Ehrenberger, T.; Zanotelli, V.; Butscheid, Y.; Escher, C.; Vitek, O.; Rinner, O.; Reiter, L. Extending the Limits of Quantitative Proteome Profiling with Data-Independent Acquisition and Application to Acetaminophen-Treated Three-Dimensional Liver Microtissues. *Mol. Cell. Proteomics* **2015**, *14* (5), 1400–1410. https://doi.org/10.1074/mcp.M114.044305.

(46) Kleiner, D. E.; Brunt, E. M.; Van Natta, M.; Behling, C.; Contos, M. J.; Cummings, O. W.; Ferrell, L. D.; Liu, Y.-C.; Torbenson, M. S.; Unalp-Arida, A.; Yeh, M.; McCullough, A. J.; Sanyal, A. J.; Nonalcoholic Steatohepatitis Clinical Research Network. Design and Validation of a Histological Scoring System for Nonalcoholic Fatty Liver Disease. *Hepatology* **2005**, *41* (6), 1313–1321. https://doi.org/10.1002/hep.20701.

(47) He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.-S. Neural Collaborative Filtering. *arXiv [cs.IR]*, 2017. https://doi.org/10.1145/3038912.3052569.

(48) Howard, J.; Gugger, S. Fastai: A Layered API for Deep Learning. *Information* **2020**, *11* (2). https://doi.org/10.3390/info11020108.

(49) Howard, J.; Gugger, S. *Deep Learning for Coders with Fastai and PyTorch: AI Applications*

Without a PhD; O'Reilly, 2020; p 582.

(50) Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.-A. *Extracting and Composing Robust Features with Denoising Autoencoders*; 2008.

(51) 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*; 2010; Vol. 11, pp 3371–3408. http://jmlr.org/papers/v11/vincent10a.html.

(52) Kingma, D. P.; Welling, M. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning* **2019**, *12* (4), 307–392. https://doi.org/10.1561/2200000056.

(53) Yu, R. A Tutorial on VAEs: From Bayes' Rule to Lossless Compression. *arXiv [cs.LG]*, 2020. http://arxiv.org/abs/2006.10273.

(54) 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.

(55) Im, D. J.; Ahn, S.; Memisevic, R.; Bengio, Y. Denoising Criterion for Variational Auto-Encoding Framework. *31st AAAI Conference on Artificial Intelligence, AAAI 2017* **2015**, 2059–2065.

(56) Harris, C. R.; Millman, K. J.; van der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; Kern, R.; Picus, M.; Hoyer, S.; van Kerkwijk, M. H.; Brett, M.; Haldane, A.; del Río, J. F.; Wiebe, M.; Peterson, P.; Gérard-Marchant, P.; Sheppard, K.; Reddy, T.; Weckesser, W.; Abbasi, H.; Gohlke, C.; Oliphant, T. E. Array Programming with NumPy. *Nature* **2020**, *585* (7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2.

(57) The pandas development team. *Pandas-Dev/pandas: Pandas*; 2022. https://doi.org/10.5281/zenodo.7093122.

(58) Mc Kinney, W. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*; van der Walt, S., Millman, J., Eds.; 2010; pp 56–61.

(59) Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Others. Pytorch: An Imperative Style, High-Performance Deep Learning Library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*.

(60) Tyanova, S.; Temu, T.; Sinitcyn, P.; Carlson, A.; Hein, M. Y.; Geiger, T.; Mann, M.; Cox, J. The Perseus Computational Platform for Comprehensive Analysis of (prote)omics Data. *Nat. Methods* **2016**, *13* (9), 731–740. https://doi.org/10.1038/nmeth.3901.

(61) Vallat, R. Pingouin: Statistics in Python. *J. Open Source Softw.* **2018**, *3* (31), 1026. https://doi.org/10.21105/joss.01026.

(62) Seabold, S.; Perktold, J. Statsmodels: Econometric and Statistical Modeling with Python. In *Proceedings of the 9th Python in Science Conference*; Austin, TX, 2010; Vol. 57, pp 10–25080.

(63) 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.* **1995**, *57* (1), 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x.

(64) Mazzanti, S. Mrmr-Selection. *PyPI*. 2022. https://pypi.org/project/mrmr-selection/0.2.5/ (accessed 2022-09-16).

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

(66) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, É. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12* (85), 2825–2830.

(67) Perez-Riverol, Y.; Csordas, A.; Bai, J.; Bernal-Llinares, M.; Hewapathirana, S.; Kundu, D. J.; Inuganti, A.; Griss, J.; Mayer, G.; Eisenacher, M.; Pérez, E.; Uszkoreit, J.; Pfeuffer, J.; Sachsenberg, T.; Yilmaz, Ş.; Tiwary, S.; Cox, J.; Audain, E.; Walzer, M.; Jarnuczak, A. F.;

Ternent, T.; Brazma, A.; Vizcaíno, J. A. The PRIDE Database and Related Tools and Resources in 2019: Improving Support for Quantification Data. *Nucleic Acids Res.* **2019**, *47* (D1), D442–D450. https://doi.org/10.1093/nar/gky1106.