Expanding the proteomics and metabolomics toolkit with methods for differential expression analysis from transcriptomics

Sebastian Dohm-Hansen[1,2,3], Maria Giovanna Caruso[1,2], Sarah Nicolas[1,2], Caitriona Scaife[4], Olivia F. O'Leary[1,2], Yvonne M Nolan[1,2], Aonghus Lavelle[1,2+], Jane A English[1,3+*]

[1] Department of Anatomy and Neuroscience, University College Cork, Cork, T12 XF62, Ireland

[2] APC Microbiome Ireland, University College Cork, Cork, T12 YT20, Ireland

[3] INFANT Research Centre, University College Cork, Cork, T12 YE02, Ireland

[4] School of Biomolecular and Biomedical Science, Conway Institute, University College Dublin, Dublin, D04 HFH4, Ireland

+ equal contribution

*Email: jane.english@ucc.ie (Jane A. English)

Abstract

With the increasing adoption of discovery -omics in the life sciences, a large number of analysis tools for differential expression analysis (DEA) have been introduced over the years. While such tools tend to be developed with one particular -omics modality in mind, they can often be applied across technologies to solve common issues. This is particularly the case when -omics data share statistical and distributional properties. Herein, we showcase how tools originally developed for transcriptomics analysis are especially well-suited to solving problems in discovery proteomics and metabolomics. Using data from our own experimental work as examples of real-world implementation, we demonstrate how these methods can be used to tackle common DEA issues, such as variable sample quality, hidden batch effects, normalization, and small sample size. We

believe this can be useful to novices and seasoned practitioners alike by expanding their toolkits. As multi-omic and integrative analyses become commonplace, it is especially useful to capitalize on the similarities of otherwise different -omics.

Keywords: differential expression analysis, bioinformatics, proteomics, metabolomics, transcriptomics, multiomics, normalization, sample quality, batch effects

Introduction

There is a saying among some machine learning practitioners, "all data is the same [sic]". Often aimed at beginners in the field, the proverb highlights how the source of tabular data (or data that can be rendered tabular) is inconsequential to any statistical model. Thus, while humans might consider the number of medications taken by a patient, the pixel intensities in an image, or the frequency of certain words in an email fundamentally different "types" of data, they are ultimately reducible to mere numbers in a matrix. What matters are the distributional characteristics of the data and whether or not they dovetail with the core assumptions of a given statistical model.

Similarly, in bioinformatics, there is the number of counts mapping to a gene, the peak area of a metabolite feature, or the label-free quantification (LFQ) intensity of a protein. While these data originate from different technologies and display some inherent distributional differences (e.g. the frequency and reason for data missingness), they also share statistical properties (e.g. heteroscedasticity and roughly quadratic mean-variance relationships following log-transformation; **Figure 1**) [1–3]. Importantly, such similarities can be exploited in differential expression analyses (DEA) to allow researchers to leverage computational tools originally developed for one technology (e.g. microarray or RNA-sequencing) in the context of another (e.g. mass spectrometry-based omics). This can be particularly useful when tools are available to address common issues across -omics modalities

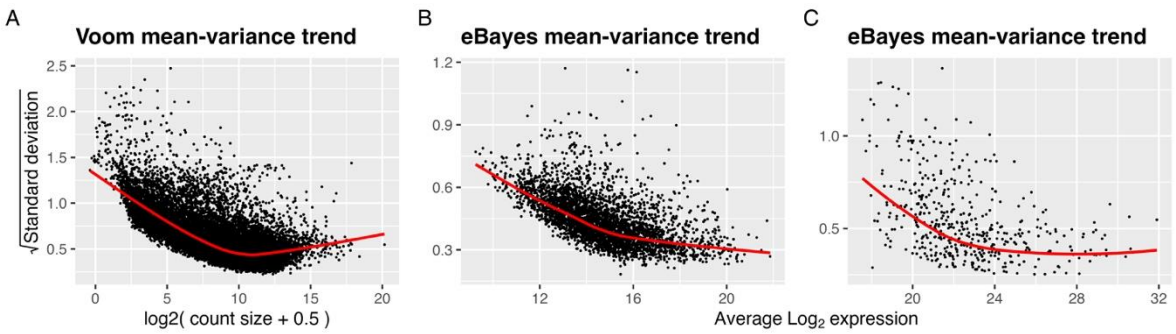


Figure 1. Mean-variance relationships of different -omics features. A) Gene-level RNA-sequencing counts on the counts per million scale (CPM) following "*limma-voom*" transformation. B) Protein label-free quantification (LFQ) intensity normalized with "*MaxLFQ*". C) Metabolomic feature peak areas following variance stabilizing normalization (*vsn*). The trends (red lines) in B) and C) were estimated by *limma-trend*. All data are expressed on the $\log_2$ scale.

With the increasingly widespread adoption of discovery -omics in the life- and medical-sciences, the number of computational tools for differential expression analysis (DEA) has steadily increased. While innovation and improvements upon previous pipelines is inarguably beneficial for bioinformaticians, there may be reproducibility- and comparability-associated issues to using increasingly disparate methods of analysis. Often, benchmarking is performed on simplified models (such as spike-ins) or limited to just a few datasets, which does not necessarily generalize to more complex experiments, such as human or animal studies [4]. Moreover, as the ground truth is rarely available to us, it can be difficult to identify a valid benchmark in the first place [4]. Additionally, for students and novices, the thicket of computational tools can be overwhelming and confusing. For seasoned practitioners, tools may be available to solve common issues, but they may not necessarily be aware of them as they lie outside their usual domain of expertise (**Table 1**).

69

Table 1. Common issues in differential expression analysis and possible remedies				
Issue	**Example source**	**Consequence(s)**	**Example tool**	**Mechanism**
Variable sample quality	Sample preparation	Low power, noise	arrayWeights (*limma*)	Empirical sample quality weights based on average mean-variance trend
Hidden batch effects	Un-/observed technical variation	Conservative *p*-values, low power	*sva*	Estimate latent factors based on variance structures unrelated to experimental design and known covariates
Normalization	Biological and technical variation	Low power	*vsn*	Linear transformation based on sample mean-variance trends
Small sample size	Resource constraints	Low power	*limma*	Borrowing information across samples to improve variance estimates with empirical, Bayesian prior

70

Herein, we want to expand the analyst's toolkit with well-established R-based tools from the transcriptomics literature that we have found useful when conducting DEA in mass spectrometry-based proteomics and metabolomics. We want to demonstrate how the same analysis approach can readily be used across -omics modalities (**Figure 2**), using data from our animal experimental work as real-world cases of implementation (**Supplementary table 1**). We draw on three label-free datasets with known issues and showcase how we addressed them. Common methodological considerations in MS-based -omics, such as data missingness and imputation, are also covered. We focus on processing steps downstream initial feature detection and annotation (which are widely different in metabolomics and proteomics and beyond the scope of this article) at the point where a matrix of features and their peak areas (or other abundance measures) is available. Finally, although we rely exclusively on R for analyses, we highlight in our conclusion how these tools can be integrated with other popular software and languages, like Perseus and Python. While all -omics might not be the same, their similarities are worth exploiting.

DEA pipeline example
- Feature data frame
- Experimental metadata

pgroup_df

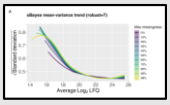
protein_ids	S_1	S_2	...	S_M
$protein_1$	$X_{1,1}$	$X_{1,2}$	...	$X_{1,m}$
$protein_2$	$X_{2,1}$	0	...	0
...	...	...	...	...
$protein_n$	$X_{n,1}$	0	...	$X_{n,m}$

exp_design

sample	condition
S_1	Treatment
S_2	Control
...	...
S_m	Treatment

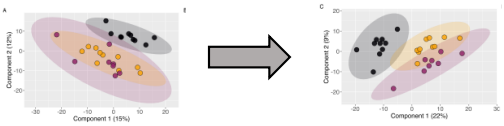
Step 1. Filtering missing values

```
# Proteomics data
# Log2 transform MaxLFQ intensity values & replace
-Inf (i.e. Log(0)) with NA (i.e. missing value)
pgroup_df <- pgroup_df %>%
  mutate(across(where(is.numeric), ~log2(.x))) %>%
  mutate(across(where(is.numeric),
~ifelse(is.infinite(.x), NA, .x)))

# Filter out protein groups with >25% missingness
in any experimental condition
pgroup_df <- pgroup_df %>%
  pivot_longer(where(is.numeric),
               names_to="sample") %>%
  left_join(exp_design) %>%
  group_by(protein_ids, condition) %>%
  mutate(is_missing=sum(is.na(value))/n()>0.25)%>%
  group_by(protein_ids) %>%
  filter(sum(is_missing)==0) %>%
  select(-is_missing, -condition) %>%
  ungroup() %>%
  pivot_wider(names_from="sample")

# LFQ matrix for SVA & limma
lfq_matrix <- pgroup_df %>%
  column_to_rownames("protein_ids") %>%
  select(where(is.numeric)) %>%
  as.matrix()
```



Step 2. Feature normalization

```
# Metabolomics data
# Feature normalization of filtered data using VSN
with default outlier tolerance = 10%
lfq_matrix <- justvsn(lfq_matrix, lts.quantile = 0.9)
```



Step 3. Surrogate variables

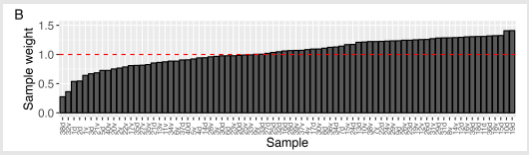
```
# SVA
# Estimate surrogate variables based on LFQ
matrix to adjust for hidden expression
heterogeneity
# full model matrix - including both
adjustment & experimental variables
mod <- model.matrix(~condition, exp_design)

# The null model contains only the adjustment
variables or intercept
mod0 <- model.matrix(~1, exp_design)

# Estimate number of surrogate variables to
include with recommended 'leek' method
num_sv <- num.sv(na.omit(lfq_matrix), mod,
method="leek")

# Run sva with the suggested number of SVs and
LFQ matrix with NA values omitted
svobj = sva(na.omit(lfq_matrix), mod, mod0,
n.sv=num_sv)

# Include surrogate variables in exp_design as
'sv_1', 'sv_2', etc.
exp_design <- cbind(exp_design, svobj$sv) %>%
  rename_with(.cols=where(is.double),
              ~paste0("sv_", .x))
```

Step 4. Design matrix

```
# Experimental design formula including
experimental condition and surrogate variables,
excluding intercept
design_formula <-as.formula(paste0("0+condition+",
                                   exp_design %>%
select(matches("sv_")) %>% colnames(),
                                   collapse="+"
                                  )
                            )

# limma design matrix for DEA
design_matrix <- model.matrix(design_formula,
                              exp_design)

# Contrast matrix
contrast_matrix <-
makeContrasts(Contrast.A=treatment-control,
              levels=design_matrix)
```

Step 5. Sample weights

```
# Sample quality weights to adjust for
differential noise levels across samples
weights <- arrayWeights(lfq_matrix, design,
                        method="reml")
```



Step 6. *limma*-trend

```
# limma-trend with robust estimation and sample quality weights
vfit <- lmFit(lfq_matrix, design=design_matrix, weights=weights)
vfit <- contrasts.fit(vfit, contrast_matrix)
efit <- eBayes(vfit, trend=T, robust=T)
```

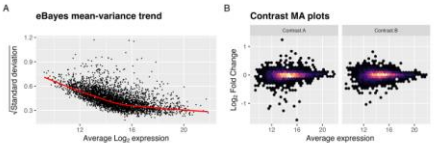


Figure 2. Overview of differential expression analysis pipeline for proteomics and metabolomics data using R-based packages. Starting with a data frame containing intensity values of features (e.g. peak areas of metabolites or label-free quantification of proteins) for each sample (*n* features by *m* samples) and a data frame with experimental metadata (i.e. mapping samples to condition; *m* samples by 2), features with a given fraction of missing values in each condition (here >25%) are filtered out (Step 1). If the data are raw intensity values, the remaining features can be transformed by variance stabilizing normalization (Step 2). The "*sva*" package can then be used to estimate surrogate variables that will be included as model covariates to account for latent noise (Step 3). A design matrix and contrast matrix are made to define the comparisons during differential expression analysis; this will include the estimated surrogate variables as covariates (Step 4). Using this design, the "arrayWeights" function in "*limma*" can be used to estimate sample quality weights that account for uneven sample quality (Step 5). Finally, differential expression analysis is conducted using *limma-trend* with or without robust estimation to account for hypo-/hypervariable features (Step 6).

Differential expression analysis with limma

In the field of bulk RNA-sequencing analysis, several gold standard tools exist for data pre-processing and downstream DEA, such as *limma* [5], *edgeR* [6], and *DEseq2* [7], all of which are based on rigorous statistical models, years of benchmarking, and widespread adoption by the bioinformatics community. In systematic comparisons, all three libraries tend to perform comparably and favourably relative to other packages [8,9]. However, they diverge in how they address the inherent heteroscedasticity of RNA-sequencing data. Sequencing counts are understood to be distributed according to the negative binomial distribution [10], which *edgeR* [6] and

DEseq2 [7] use as a foundational assumption. *limma* on the other hand was originally developed to analyze RNA microarray with linear models [11] and only later expanded to accommodate RNA-sequencing data [3]. Through its "*trend*" and "*voom*" methods, *limma* instead models the gene-level mean-variance relationship (i.e. heteroscedasticity) with a LOWESS trend and either incorporates it as a prior in its Bayesian moderated *t*-tests (*limma-trend*), called "*eBayes*" [11], or in the form of inverse precision weights to eliminate it (*limma-voom*) [12]. This approach allows *limma* to leverage the flexibility and power of normal-based statistics during DEA, such as the ability to incorporate individual sample quality weights or include random effects to accommodate repeated-measures or nested designs, neither of which is possible with count distribution-based models [3]. Moreover, compared to traditional statistical testing, the Bayesian prior used by *limma* increases statistical power, especially in designs with few samples, by effectively "borrowing" information across genes in all samples rather than only those being compared (i.e. as in a standard *t*-test). Thus, in trading off the negative binomial assumption, *limma* could leverage the power and flexibility of normal-based statistics.

As noted previously, mass spectrometry-based (MS) -omics, such as bottom-up proteomics, often display feature-wise mean-variance relationships that are similar to those seen in transcriptomics. Indeed, LFQ distributions, peptide counts, and peptide-spectrum matches (PSM) all display inherent heteroscedasticity (**Figure 3**). *limma* has since been repurposed to model these relationships in proteomics data. The R package *DEqMS* [2] is built as a wrapper around *limma* and uses its assumption-free LOESS to fit protein-level variances against peptide or PSM counts. The trend is then included in the downstream, moderated *t*-tests (called "*spectraCounteBayes*") for protein-level variance shrinkage. Interestingly, however, this approach performs comparably to simply using *limma-trend* to fit protein variances against protein intensity,

especially as sample size and biological variance increase [2]. This is likely due to the similar mean-variance distributions of protein-level LFQ data and peptide counts (**Figure 3**). Thus, in cases where peptide or PSM counts are not available, or in experiments with high biological variance (i.e. most human and animal studies), *limma-trend* would be expected to perform as well as *DEqMS*. In this regard, it is worth noting that the R package "*DEP*" [13] for proteomics runs *limma* under the hood to conduct DEA. Moreover, *limma* is an available option within the popular *Perseus* software for computational proteomics [14,15].

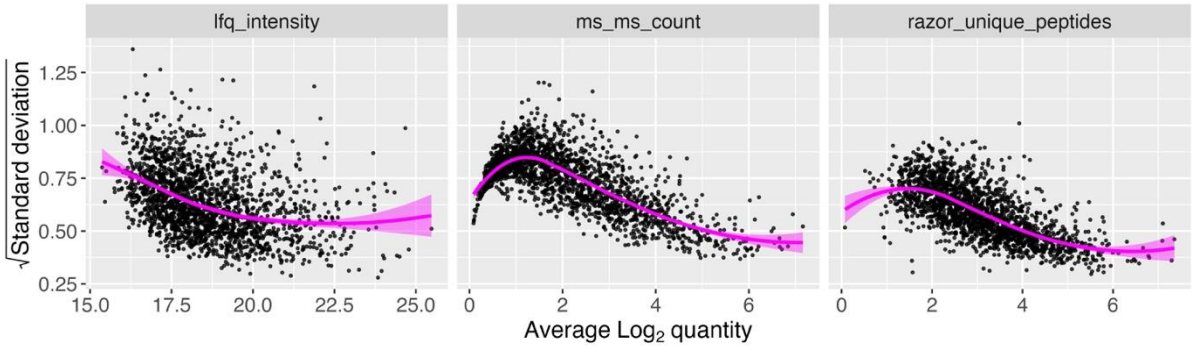


Figure 3. Mean-variance relationships of label-free proteomics data. Protein label-free quantification (LFQ) intensity (*left panel*), ms/ms counts (*middle panel*), and razor+unqiue peptide counts (*right panel*) all display similar mean-variance relationships. Proteins have been filtered for maximum 25% missingness prior to plotting. Magenta trend lines represent models fitted by LOESS to the data. All data are expressed on the $\log_2$ scale.

limma has also been used to conduct DEA in untargeted metabolomics studies, including by the authors [16,17]. Indeed, the mean-variance relationship of feature-level peak area (**Figure 1C**) can be similarly exploited by *limma-trend*. Importantly, in all these uses cases, *limma* is superior compared to traditional parametric statistics (i.e. Student's *t*-test and ANOVA) in terms of power

and false discovery rate (FDR) control [2,3]. In summary, its assumption-free modelling of mean-variance relationships makes it flexible and readily usable across -omics modalities. Finally, as part of the Bioconductor library of R packages [18], *limma* has extensive documentation and a rigorous code base.

Data missingness and imputation

A critical initial step in any analysis pipeline is to assess the completeness of the dataset. Due to the inherently stochastic nature of MS acquisition, datasets in proteomics [19] and metabolomics [20] often display varying levels of missingness. This is particularly the case with Data-Dependent Acquisition (DDA), which initially targets a wide mass/charge (m/z) spectrum in which the top *N* most abundant precursors are selected for fragmentation and MS/MS acquisition. In comparison, with recent developments, Data-Independent Acquisition (DIA) tends to produce fewer missing values [21]. Thus, a degree of randomness is introduced due in part to acquisition mode on the MS, resulting in what is termed missingness at random (MAR); that is, features may or may not be detected independent of their characteristics or abundance [22]. As such, missing values will be randomly distributed among samples and features. Conversely, due to the competition dynamics of top-*N* precursor selection, some features will be missing systematically owing to their inherent abundance (or characteristics) in the sample, resulting in what is termed missingness not at random (MNAR) [22]. Both types of missingness are known to occur in a sample-dependent manner that can be modelled [22].

Ultimately, feature missingness can be addressed in two, non-mutually exclusive ways: filtering and imputation. Generally, filtering out lowly expressed features tends to select for more robust features and further helps to alleviate the multiple comparison problem [23]. From a statistical

point of view, it ensures there is sufficient information on which to base fold changes and mean-variance trends. Filtering can be done in one of several ways: as a percentage of all samples, as a percentage in each experimental condition, completeness in a minimum number of samples, among others. As an aside, untargeted metabolomics data usually requires additional, stringent filtering due to the high level of redundant and potentially spurious features [24].

To deal with the remaining missing values, imputation has been commonplace in MS-based -omics to improve power and because some tools (e.g. PCA) cannot handle missing values. Indeed, a fairly routine workflow in the "*Perseus*" software for computational proteomics involves imputation by random draws from a Gaussian distribution that is left-shifted and scaled relative to the overall sample intensity distribution (i.e. assuming MNAR due to low abundance; though other methods are available) [15]. Several R packages developed for MS-based -omics feature imputation, such as "*MSnbase*" [25], "*DEP*" [13], and "*ImputeLCMD*" [26]. Two commonly used methods to address MAR and MNAR, respectively, are K-nearest neighbors (KNN) and quantile regression for imputation of left-censored data (QRILC). However, as noted previously, proteomics datasets feature both types of missingness at varying levels. To address this, a hybrid imputation algorithm was developed Lazar et al. [22] and included in the R package "*ImputeLCMD*" [26]. Here, the missingness is modelled as a function of abundance (i.e. feature intensity), and a threshold is estimated below which missingness is assumed to be MNAR, while those above are assumed to be MAR. These are, then, imputed by default with QRILC and KNN (though, other methods are available). Additionally, it has been suggested that conducting such hybrid imputation stratified by experimental condition can increase power [27]. Indeed, missingness can be systematically different between conditions, so one could argue that it makes sense to "borrow" information from

each condition when imputing to improve estimates. As a caveat, it is plausible that such an approach could inflate false positives by artificially amplifying differences between conditions.

We have investigated the effects of different imputation methods on proteomic LFQ distributions and DEA results as a function of maximum allowed feature missingness in each experimental condition. We used a wide range for demonstration purposes, but it should be noted that a missingness threshold in excess of 50% would be outside the norm. As can be seen in **Figure 4**, the MNAR methods ("Perseus" and "QRILC") impute on the lower end of the distribution and significantly distort it when more than 50% missing values in each condition are allowed (note the bimodal peaks that emerge in **Figure 4**). Conversely KNN and the hybrid imputation by experimental condition ("Hybrid") methods better preserve the overall distribution even in the face of high feature missingness. Indeed, while a method like KNN will be able to impute some MNAR values correctly, solely relying on a left-censored approach will systematically underperform on non-MNAR values [22].

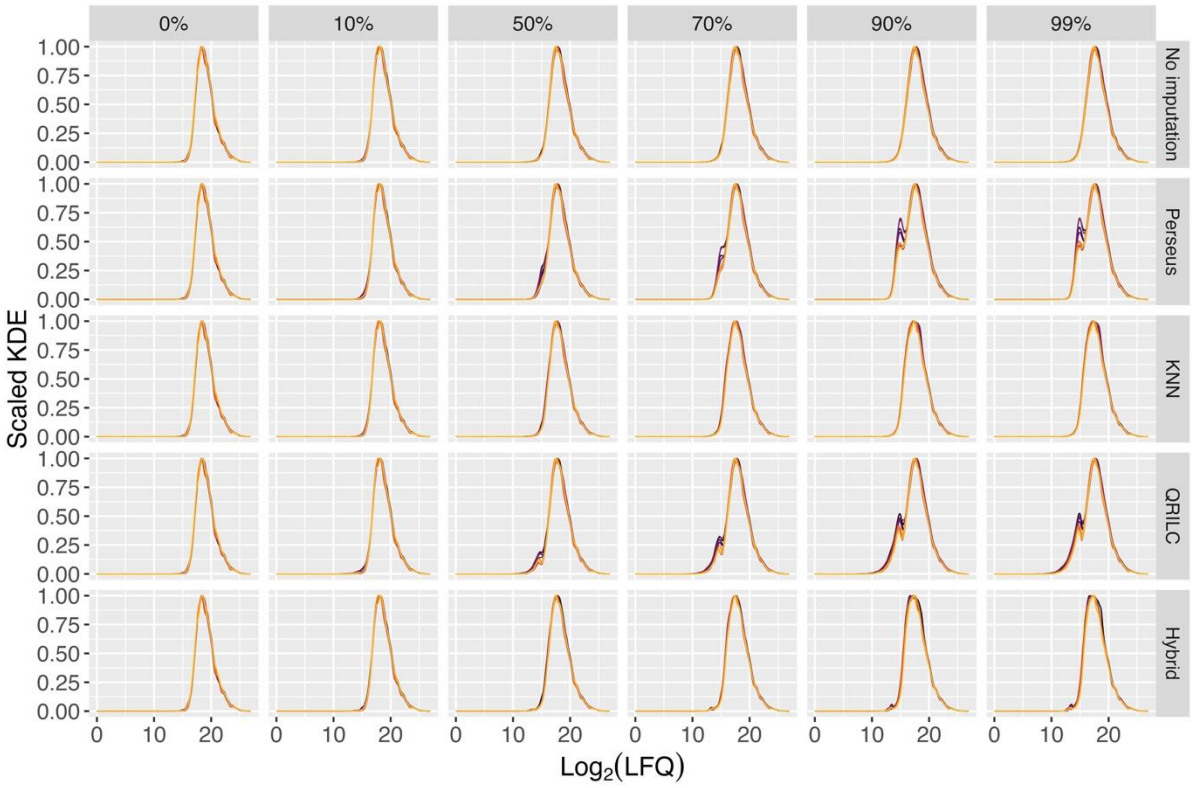


Figure 4. Effects of imputation and missingness filtering threshold on feature intensity distributions. Proteins were filtered at various thresholds (0-99%) of maximum allowed missingness in each experimental condition (represented by differently coloured distributions). Remaining missing values were then imputed by KNN, QRILC, or the Perseus method using the package "DEP". Alternatively, the hybrid imputation method from the "*ImputeLCMD*" package was run separately for each experimental condition with parameters KNN and QRILC for MAR and MNAR imputation, respectively. The raw, un-imputed data were included as a reference ("No imputation"). Distributions were approximated by Kernel Density Estimation (KDE) and scaled to range from 0-1. Filtering and imputation were conducted on label-free quantification (LFQ) intensity following $\log_2$ transformation. Each experimental condition has N=10 samples. Note the bimodal peaks that emerge with the "Perseus" and "QRILC" methods at missingness thresholds >50%.

When assessing the combined impact of missingness filtering threshold and imputation on DEA results, we generally find that the biggest differences similarly occur on the high end. Here, we assessed the number and percentage of proteins reaching FDR < 0.05 following *limma-trend* DEA as a function of imputation method and missingness threshold (same as those used above). In this example, outcomes rapidly begin diverging as the threshold is made more lenient than 25% (**Figure 5**). Notably, the hybrid imputation approach that takes experimental condition into account generates a particularly high number of significant features. While perhaps not surprising, there is a significant risk that such an implementation would differentially amplify noise across conditions, thus increasing the number of false positives. Indeed, this risk should be carefully considered when using experimental covariates to inform imputation, especially in the face of high levels of feature missingness. While different imputation methods can result in widely different numbers of significant features, in the absence of ground truth, it can be hard to know which level of differential expression to expect. Indeed, more is not necessarily better, and results should be carefully evaluated. Critically, imputation should only be carried out when there is sufficient complete data to accurately model it.

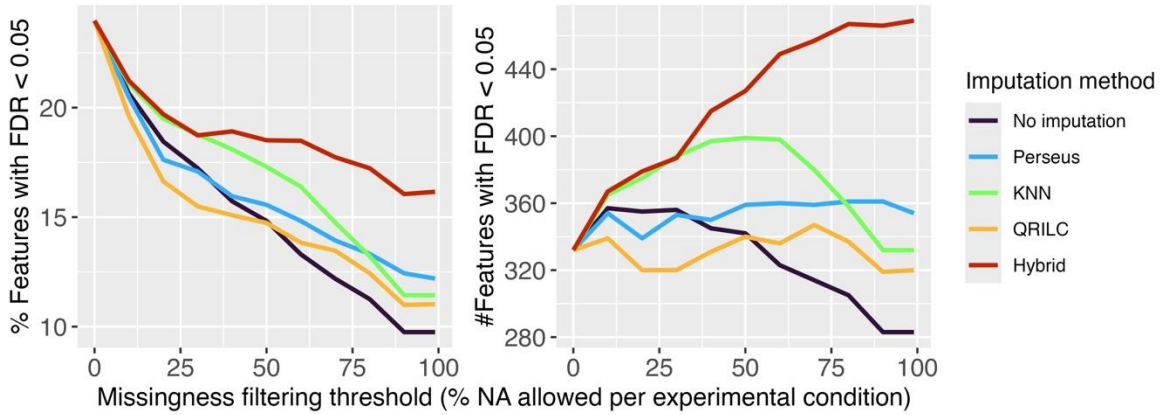


Figure 5. Effects of feature imputation and missingness filtering threshold on differential expression analysis with *limma-trend*. Proteins were filtered at various thresholds (0-99%) of maximum allowed missingness in each experimental condition, and the remaining missing values were imputed by various methods (differently coloured lines). Filtering and imputation were conducted on label-free quantification (LFQ) intensity following $\log_2$ transformation. Differential expression analysis was then carried out using *limma-trend* with the robust setting (to account for the increased variability due to the presence of features with high levels of missingness), and the number of features with a Benjamini-Hochberg-based false discovery rate (FDR) < 0.05 were reported either as a percentage of all protein groups (*left panel*) or as absolute numbers of proteins (*right panel*).

Ultimately, whether or not to impute, and if so, by which method, is up to the researcher and depends on the dataset (proteomics versus metabolomics), acquisition method (e.g. DDA versus DIA), and research question (e.g. DEA versus profiling), as is the filtering threshold and method. No universal best imputation strategy exists, and sometimes, omitting imputation altogether is optimal [28]. We generally opt not to impute and tend to filter such that every feature has at least 75% valid values in each experimental condition to ensure reliable fold change estimates. We additionally investigate the impact of different thresholds on the resulting mean-variance trend and DEA fold-changes (**Figure 6**). Fortunately, packages like *limma* handle missing data, and more recent alternatives exist to methods that otherwise do not accept missing values, like principal component analysis (PCA) [29].

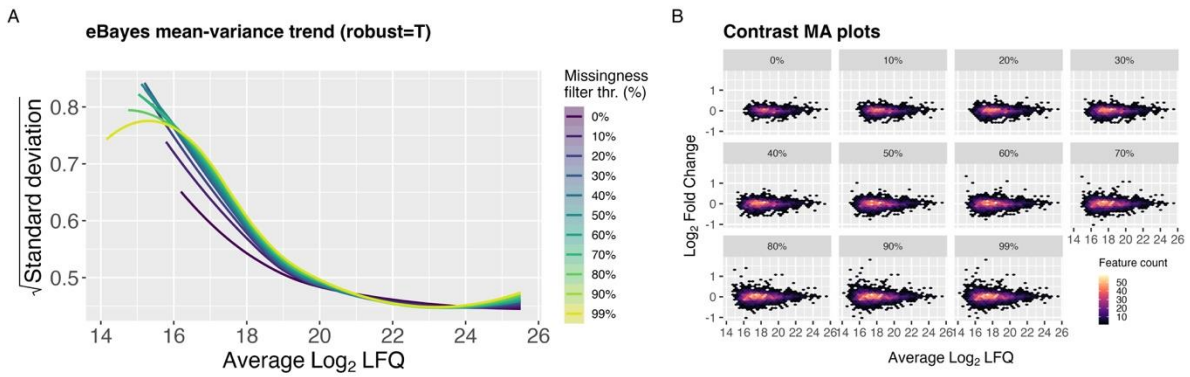


Figure 6. Effects of missingness filtering threshold on mean-variance trend and fold-change distributions. Proteins were filtered at various thresholds (0-99%) of maximum allowed missingness in each experimental condition. Filtering was conducted on label-free quantification (LFQ) intensity following $\log_2$ transformation without imputation. Differential expression analysis was then carried out using *limma-trend* with robust estimation to account for the increased variance attributable to missingness. A) Intensity-based prior trends as a function of maximum feature missingness. B) MA plots displaying protein fold changes as a function of mean $\log_2$-transformed LFQ and maximum feature missingness.

Data normalization and transformation

Prior to quantification and DEA, mass spectrometry data will often be normalized to account for systematic differences due to sample loading and inter-run variability. In label-free proteomics, the "*MaxLFQ*" algorithm [30] (originally part of the *MaxQuant* [31] suite and later implemented in other tools, such as Fragpipe [32] and DIA-NN [33]) has been a popular means of conducting LFQ at the protein level. "*MaxLFQ*" estimates normalization factors that ensure minimum differential regulation for most proteins based on pairwise comparisons between all samples. That is, the algorithm assumes most protein group abundances are roughly equal across conditions and samples and uses this as an "internal standard". This is what we have relied on herein. A similar

approach is used by the commercial software suite Progenesis QI from Waters in the case of untargeted metabolomics, whereby scaling factors are estimated for each compound ion, assuming the abundances of most compounds do not change across samples. Finally, within the popular freeware Metaboanalyst, several normalization procedures exist to address systematic differences between samples [34].

Beyond these methods, several normalization techniques exist to address different issues in the data that obscure biological effects of interest. For instance, centering (i.e. by median or mean subtraction), scaling (e.g. z-scoring), and log-transformation of (already normalized) data can be used to remedy linear offsets, scale differences, and heteroscedasticity in the data [1]. This can be done either at the sample or feature level depending on the research question and distribution of the data. Importantly, datasets will differ in terms of technical and biological variance, presence of outliers and systematic bias, so it is advisable to trial and compare several methods on a case-by-case basis [35,36]. In systematic comparisons [37–40], noteworthy examples that tend to perform well on both proteomics and metabolomics data are cubic splines [41], quantile normalization [42], and variance stabilizing normalization (vsn) [43], which were all originally developed for microarray data.

We have compared these three methods in our metabolomics work and found that they reliably improve clustering in PCA space (**Figure 7A**) and increase the number of differentially expressed features compared to $\log_2$ transformation alone (**Figure 8**). However, it is important to note that all have drastically different impacts on sample intensity distributions (**Figure 7B**). For instance, cubic splines attempt to align sample distributions based on a target distribution, such as the arithmetic mean of feature intensities across all samples [41]. This has the effect of centering the pairwise log-ratios of all samples across the intensity range, which assumes that most features are

not differentially expressed. Relatedly, quantile normalization forces all sample intensity distributions to be identical [42]. In our experience, it is important to examine the effect of these methods on individual features. The latter method, in particular, can sometimes deflate individual feature variances or even set them to 0. In these cases, availing of *limma's* robust estimation of its mean-variance trend [44] can protect against such hypovariable outliers.

We have favoured the use of *vsn* particularly in our metabolomics work [17], and others have found similarly good performance in label-free proteomics [39,45]. The motivation for the algorithm has its roots in shared error models with analytical chemistry, such as gas chromatography and mass spectrometry [46]. Fundamentally, *vsn* seeks to address the limitations of using the standard log transformation to deal with heteroscedasticity. In short, due to the vertical asymptote at 0 in the domain of the log transformation, variances of low-abundant features tend to be inflated [1]. Instead, *vsn* uses a so-called generalized logarithmic function (glog), which contains a linear transformation to allow a positive range over zero-valued and negative intensities [43]. The parameters of the transformation are estimated empirically from the data through robust regression. Additionally, *vsn* performs an affine transformation (scaling and offset) that seeks to correct for background and multiplicative sample-wise noise. The function is such that for values >>0, glog_2 approaches $\log_2$. Critically, compared to the other two mentioned methods, we find that *vsn* tends to achieve comparable or better clustering in PCA space (i.e. by visual inspection), increased numbers of differentially expressed features, while also preserving individual feature intensity distributions. Relative log abundance ratios and median absolute deviations are quantitative measures that can also be used to assess the impact of normalization [36]. Moreover, several R packages have been developed to automatically evaluate (and suggest) normalization methods for proteomics [36,45] and metabolomics [35] datasets. It is worth noting that while we only used

metabolomics data to demonstrate these microarray-based normalization techniques, they can apply equally well to proteomics data, in line with our overarching theme of harnessing the common distributional properties of different -omics modalities to address common statistical challenges. Indeed, it is precisely because high-throughput discovery proteomics, untargeted metabolomics, and microarray data all lend themselves to similar error assumptions (i.e. additive background and multiplicative sample-specific noise, and heteroscedastic data) that tools like *vsn* can be used across all three. For instance, we have used *vsn* in combination with z-scoring to normalize (and variance stabilize) proteomics and metabolomics data for use with multi-omics tools like MOFA [47]. Additionally, most of the normalization tools and systematic studies mentioned earlier have applied the same microarray-based techniques to both proteomics and metabolomics data, often with good results [35,36,39,45].

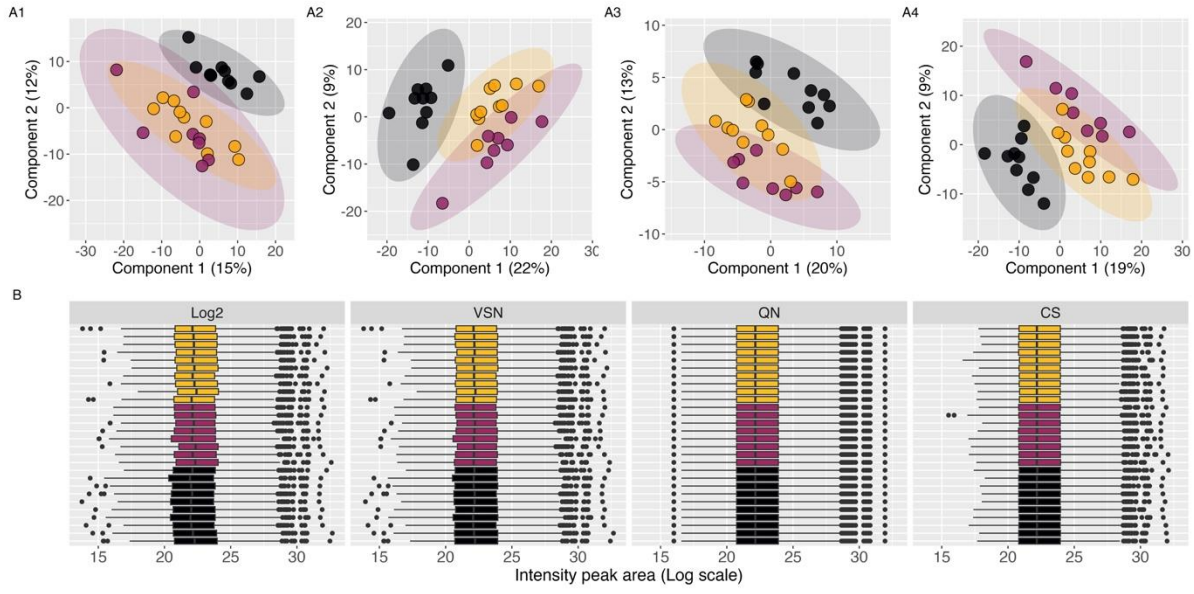


Figure 7. Effect of various normalization methods on sample clustering and intensity distributions in untargeted metabolomics data. A) Principal component analysis of feature data after $\log_2$ transformation (A1), variance stabilizing normalization, VSN (A2), quantile

normalization, QN (A3), and cubic splines normalization, CS (A4). B) Feature peak area distributions per sample across three experimental conditions (indicated by colour) following the same normalization methods as in A1-A4. Data were filtered for a maximum of 25% missingness in each condition prior to normalization.

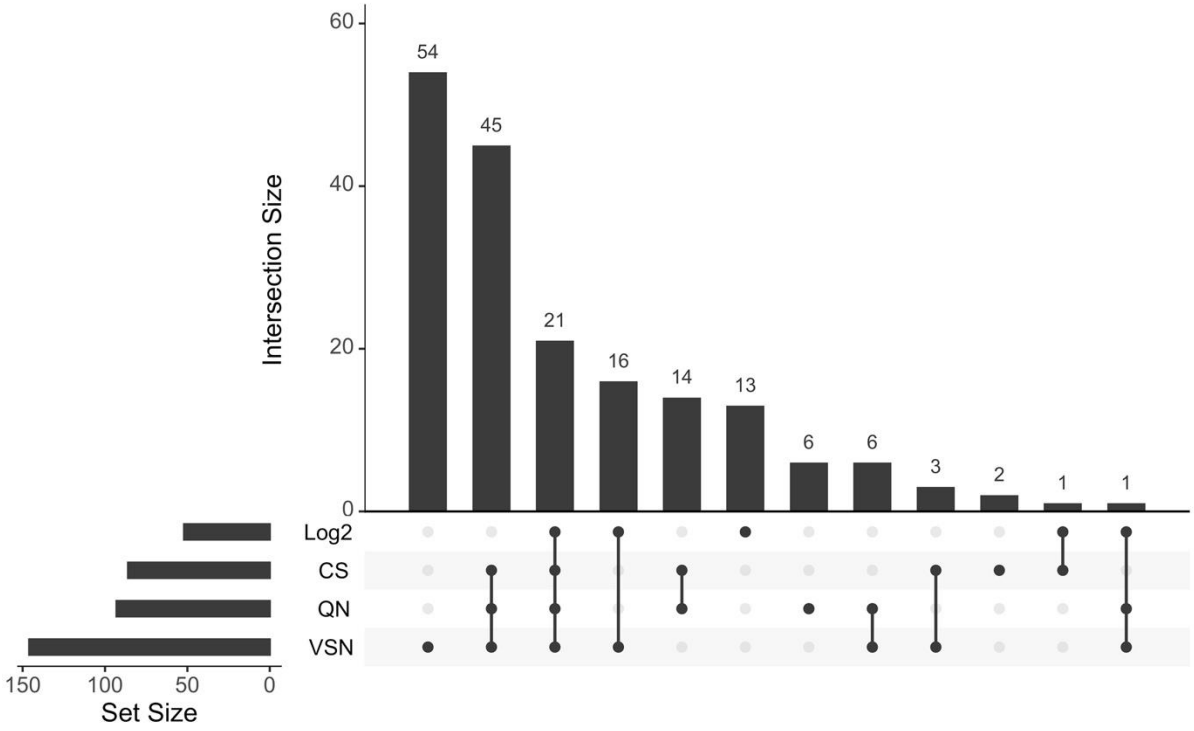


Figure 8. Effect of the various normalization methods on the number of differentially expressed features in untargeted metabolomics data. Upset plot indicating the number of features with a Benjamini-Hochberg-based false discovery rate (FDR) <0.05 following differential expression analysis with *limma-trend* with robust estimation. Features were filtered to allow a maximum missingness of 25% in each of three experimental conditions. Peak areas were then transformed with one of four methods: $\log_2$, variance stabilizing normalization (VSN), quantile normalization (QN), or cubic splines (CS). The plot indicates the total number of features with

FDR <0.05 for each method (horizontal bar chart) and the number of common features across methods (vertical bar chart). NB: data are the same as those referenced in Figure 7.

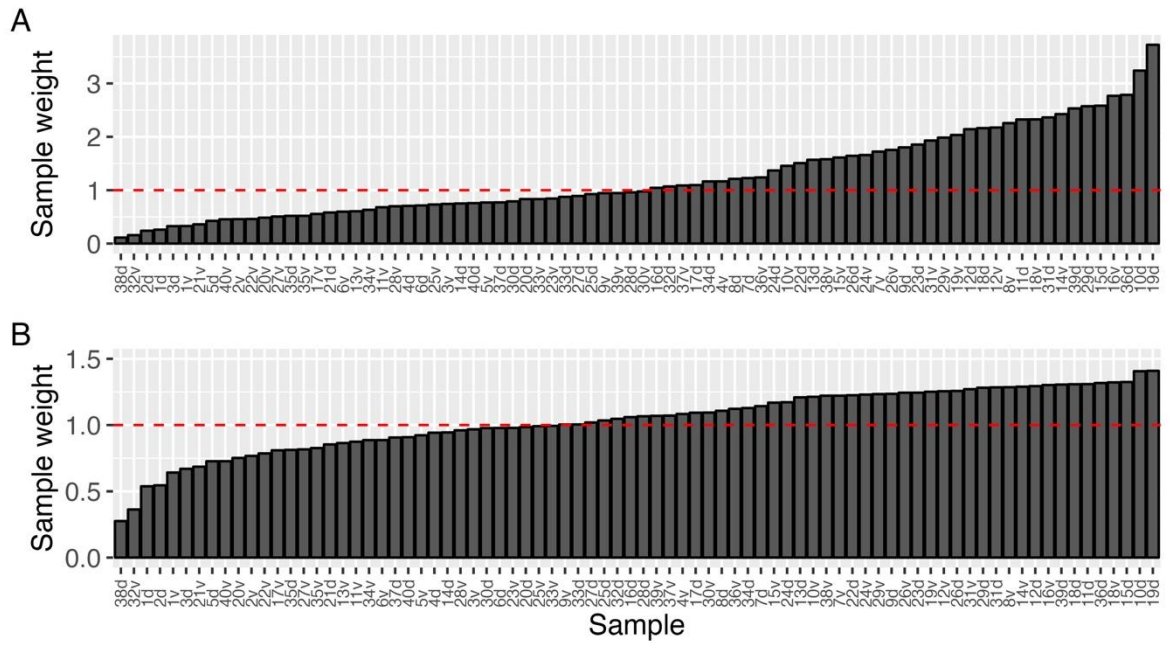
Improving power with sample quality weights

The degree of biological variance in a dataset has significant impacts on its mean-variance trend and resulting DEA fold changes [12]. Especially in animal and human studies, uneven sample quality is a common issue. This can be due to systematic errors, such as batch effects, or more random sources, such as human error during sample preparation. As noted by Liu et al. [48], this frequently leads to the dilemma of whether or not to retain noisy samples, either of which can result in lower power for different reasons.

In the transcriptomics literature, this issue has been mitigated by the advent of empirical sample quality weights. Originally developed for microarray studies, *limma's* so-called "arrayWeights" function leverages the fundamental mean-variance trend but allows each gene to not only have its own variance factor but also to depend on a sample-specific (originally array-specific) variance factor [49]. The latter is converted into a weight that reflects the degree to which a sample is more or less variable than the average sample. The weight is then used to up- or down-weight that sample accordingly during DEA. The weights can be calculated individually for all samples or in "blocks" based on the experimental design. Critically, array weights have been found to reflect actual sample quality, such as the degree of RNA degradation during benchmarking [48].

In our proteomics and metabolomics work, sample quality weights have often been indispensable. As noted earlier, animal and human samples tend to suffer from uneven quality (e.g. due to variable preparation), which can mask actual differential expression. As an example, in one of our studies, we surgically isolated dorsal and ventral sections of hippocampal dentate gyri in

rats and carried out DDA-based proteomics. Notably, the "arrayWeights" function in *limma* accurately down-weighted two samples that were expected to be of low quality due to technical issues during sample preparation (see samples "38d" and "32v" in **Figure 9**). We have since implemented sample quality weights routinely for all human and animal -omics studies. On a technical note, while the default settings of "arrayWeights" often work well (**Figure 9A**), we have occasionally found it useful to tweak its "prior.n" parameter. In short, this can be thought of as adding in a number of artificial features for which all samples are equally variable. Thus, the higher the number, the more the sample quality weights are squeezed towards equality (i.e. the prior). Depending on "prior.n", the range around equality (i.e. 1) and, thus, the maximum magnitude of weighting varies. In some datasets, we have found more stable performance (i.e. on downstream DEA and pathway analysis) by forcing the weights to be roughly symmetrical about 1 (**Figure 9B**) by setting "prior.n" equal to the number of features in the dataset. This is more conservative and mainly seeks to down-weight low-quality samples as opposed to strongly up-weight good-quality samples.



395

396 **Figure 9. Sample quality weights with *limma's* "arrayWeights" function.** Filtered and log_2-
397 transformed protein label-free quantification (LFQ) data from *N*=80 samples were input into
398 *limma's* "arrayWeights" function with either default parameters (A) or "prior.n" set to be equal to
399 the number of protein group features (B). The dotted red line represents equality. Samples with
400 weights >1 are less variable than the average sample and are up-weighted, whereas samples with
401 weights <1 are more variable and are down-weighted. In the figure, samples "38d" and "32v"
402 (which have the lowest weights) were known beforehand to be of low quality due to sample
403 preparation issues.

404

405 **Removing unwanted sources of variance with surrogate variables**

406 Sometimes, the source and structure of confounding variation within a dataset is unknown to the
407 researcher (i.e. latent) and, therefore, cannot be modelled and included as a covariate in the DEA
408 [50]. This noise, be it biological or technical in origin, can obscure effects of interest and compromise
409 power, introduce unwanted statistical dependency, and result in a lack of significant tests, or

conservative *p*-value distributions [50,51]. It presents itself as common (correlated) patterns of differential expression among many features, unrelated to the primary (i.e. independent) variables of interest. To account for such as artefacts, initially in microarray data, Leek and Storey [50,52] developed "surrogate variable analysis" (SVA), which seeks not to model the true underlying factors but rather latent variables that capture the structure of the noise after the primary variable has been accounted for. The algorithm works by conducting singular value decomposition (similar to a PCA) on the residuals of the data after the effects of all known covariates have been regressed out to capture any remaining variance structures. Features accounting for this variance are used to construct latent factors, called surrogate variables, which can then be included as covariates in the regular DEA model.

This approach has been found to significantly increase power, accuracy, and reproducibility by taking unwanted variation out of the data [50]. Indeed, we have been able to account for unmeasured sources of variation that would otherwise obscure biological effects (**Figure 10**). For instance, in an independent validation of the aforementioned brain proteomics study, we found strong evidence of latent noise that we could not account for (**Figure 10A,B**), which compromised DEA. Using "*sva*", we were able to unveil the expected biological variation (**Figure 10C,D**) and significantly increase the number of differentially expressed features overlapping with those seen in the original discovery dataset. We have since incorporated it in both our proteomics and metabolomics work. It should be noted that "*sva*" does not support missing values, and in these cases, we have occasionally found hybrid imputation useful. While "*sva*" rarely features in the mass spectrometry-based -omics literature, it has been suggested before to address batch effects in proteomics, but as with all other tools, its utility has to be evaluated on a case-by-case basis [53].

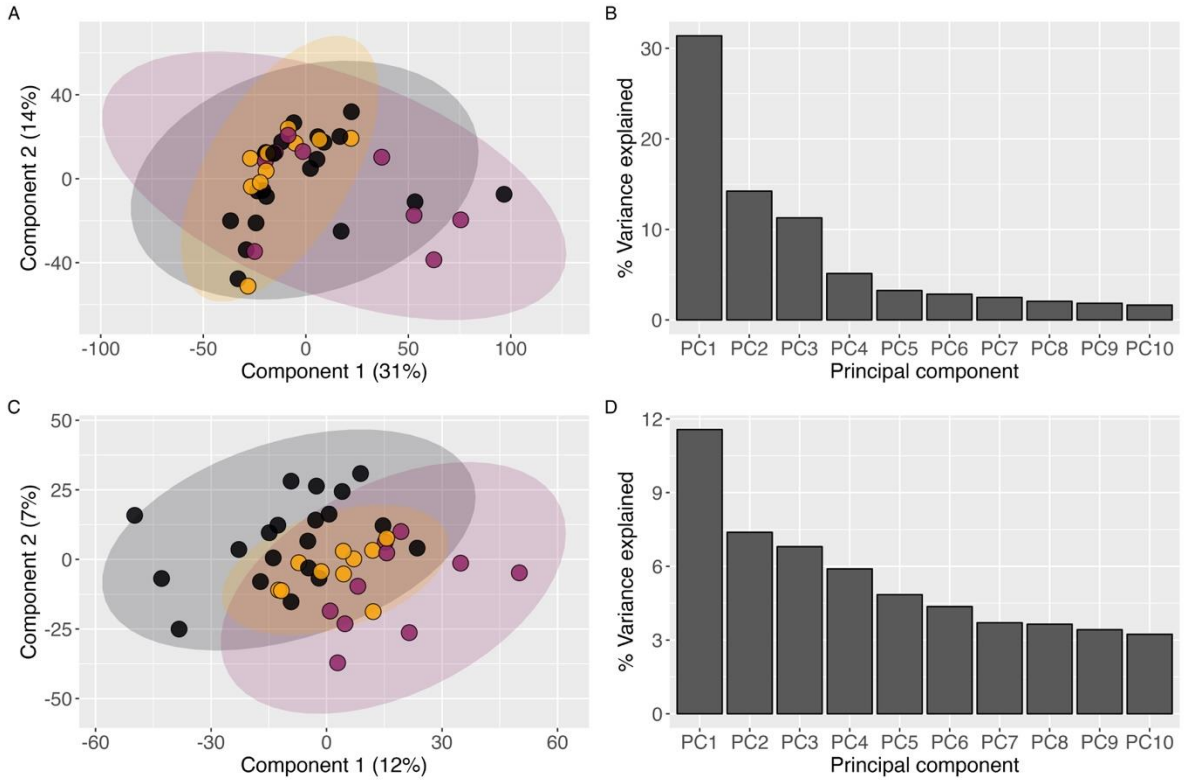


Figure 10. Surrogate variable analysis (SVA) unveils variation of interest. Principal component analysis of *N*=40 samples across three experimental conditions (represented by the different colours) based on unadjusted label-free quantification (LFQ) intensity (A,B) or residuals after regressing out three surrogate variables estimated with "*sva*" (C,D). Proteins were filtered for missingness and $\log_2$-transformed prior to PCA with "*prcomp*" (data centered and scaled). The first ten principal components were returned. Note the improved clustering by experimental condition in (A) compared to (C).

Conclusion

This article has highlighted computational tools from the transcriptomics literature that we have found to improve DEA results in discovery proteomics and metabolomics. Using all of them

446 combined, we have consistently been able to improve our power to detect differentially expressed
447 features, reduce technical variance, and improve reproducibility. Ultimately, our argument for
448 employing these methods across different -omics modalities is that they assume statistical
449 properties that are common to high-throughput datasets such as proteomics, metabolomics, and
450 transcriptomics. In this study we rely exclusively on R for analysis; however, all of the
451 abovementioned packages can in principle be employed in user-friendly software that supports R-
452 based plug-ins, such as *Perseus* (see REF [14] for a tutorial). Additionally, we have developed a
453 Python package for proteomics and metabolomics DEA, called DEPy (available as
454 "*summarizedpy*" on PYPi), that allows users to leverage all of the abovementioned R-based tools
455 (*limma-trend*, *vsn*, *sva*, *arrayWeights, ImputeLCMD*) inside Python. This is done by calling R
456 under the hood in a carefully isolated conda environment. Finally, Python users can also call R
457 natively using libraries like *rpy2*.

458 It is important to stress, however, that we do not prescribe the use of these transcriptomics
459 tools by default in all cases. Rather, we proffer them to expand the analyst's toolkit with well-
460 established and documented methods. This approach will be particularly useful in high-variance
461 datasets that display uneven sample quality and (hidden) batch effects, such as in human or animal
462 studies. Conversely, in highly controlled experiments with low technical variance, many of these
463 steps can likely be omitted. Another benefit to this approach is that DEA pipelines can be
464 standardized, and thus, comparisons across datasets may be facilitated.

465 In conclusion, leveraging the statistical similarities of different -omics modalities can open
466 up the number of available tools for DEA. There are decades' worth of tried and trusted methods
467 in the transcriptomics literature that we believe are equally useful in the proteomics and

metabolomics fields. Much like in machine learning, it is not the origin of the data that matters, but their distributional properties.

CRediT authorship contribution statement

SDH: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing - original draft, Writing - review & editing, Visualization. MGC: Investigation. SN: Investigation. CS: Software, Methodology, Formal analysis, Resources, Data Curation. OOL: Supervision, Conceptualization, Project administration. YN: Supervision, Conceptualization, Project administration, Funding acquisition, Writing - review & editing. AL: Supervision, Writing - review & editing, Conceptualization. JE: Supervision, Writing - review & editing, Conceptualization, Project administration.

Acknowledgments

YN is a recipient of a Science Foundation Ireland Investigator Award (SFI/FFP/6820), which funds SD-H. JAE is funded by the Irish Health Research Board (HRB EIA-2017-023).

Data availability

The raw MS proteomics data have been deposited to the ProteomeExchange Consortium via the MassIVE partner repository with the dataset identifier PXD066617. Additionally, the processed metabolomics data have been made available as part of the supporting information below.

Supporting information

- Supplementary Table 1: Experimental, technical, and analytical details of showcased datasets (.docx).
- Supplementary Table 2: Processed untargeted metabolomics peak area data (.xlsx).

References

(1) van den Berg, R. A.; Hoefsloot, H. C. J.; Westerhuis, J. A.; Smilde, A. K.; van der Werf, M. J. Centering, Scaling, and Transformations: Improving the Biological Information Content of Metabolomics Data. *BMC Genomics* **2006**, *7*, 142. https://doi.org/10.1186/1471-2164-7-142.

(2) Zhu, Y.; Orre, L. M.; Zhou Tran, Y.; Mermelekas, G.; Johansson, H. J.; Malyutina, A.; Anders, S.; Lehtiö, J. DEqMS: A Method for Accurate Variance Estimation in Differential Protein Expression Analysis. *Mol Cell Proteomics* **2020**, *19* (6), 1047–1057. https://doi.org/10.1074/mcp.TIR119.001646.

(3) Law, C. W.; Chen, Y.; Shi, W.; Smyth, G. K. Voom: Precision Weights Unlock Linear Model Analysis Tools for RNA-Seq Read Counts. *Genome Biol* **2014**, *15* (2), R29. https://doi.org/10.1186/gb-2014-15-2-r29.

(4) Brooks, T. G.; Lahens, N. F.; Mrčela, A.; Grant, G. R. Challenges and Best Practices in Omics Benchmarking. *Nat Rev Genet* **2024**, *25* (5), 326–339. https://doi.org/10.1038/s41576-023-00679-6.

(5) Ritchie, M. E.; Phipson, B.; Wu, D.; Hu, Y.; Law, C. W.; Shi, W.; Smyth, G. K. Limma Powers Differential Expression Analyses for RNA-Sequencing and Microarray Studies. *Nucleic Acids Res* **2015**, *43* (7), e47. https://doi.org/10.1093/nar/gkv007.

(6) Robinson, M. D.; McCarthy, D. J.; Smyth, G. K. edgeR: A Bioconductor Package for Differential Expression Analysis of Digital Gene Expression Data. *Bioinformatics* **2010**, *26* (1), 139–140. https://doi.org/10.1093/bioinformatics/btp616.

(7) Love, M. I.; Huber, W.; Anders, S. Moderated Estimation of Fold Change and Dispersion for RNA-Seq Data with DESeq2. *Genome Biology* **2014**, *15* (12), 550. https://doi.org/10.1186/s13059-014-0550-8.

(8) Seyednasrollah, F.; Laiho, A.; Elo, L. L. Comparison of Software Packages for Detecting Differential Expression in RNA-Seq Studies. *Brief Bioinform* **2015**, *16* (1), 59–70. https://doi.org/10.1093/bib/bbt086.

(9) Corchete, L. A.; Rojas, E. A.; Alonso-López, D.; De Las Rivas, J.; Gutiérrez, N. C.; Burguillo, F. J. Systematic Comparison and Assessment of RNA-Seq Procedures for Gene Expression Quantitative Analysis. *Sci Rep* **2020**, *10* (1), 19737. https://doi.org/10.1038/s41598-020-76881-x.

(10) Gierliński, M.; Cole, C.; Schofield, P.; Schurch, N. J.; Sherstnev, A.; Singh, V.; Wrobel, N.; Gharbi, K.; Simpson, G.; Owen-Hughes, T.; Blaxter, M.; Barton, G. J. Statistical Models for RNA-Seq Data Derived from a Two-Condition 48-Replicate Experiment. *Bioinformatics* **2015**, *31* (22), 3625–3630. https://doi.org/10.1093/bioinformatics/btv425.

(11) Smyth, G. K. Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments. *Stat Appl Genet Mol Biol* **2004**, *3*, Article3. https://doi.org/10.2202/1544-6115.1027.

(12) Law, C. W.; Chen, Y.; Shi, W.; Smyth, G. K. Voom: Precision Weights Unlock Linear Model Analysis Tools for RNA-Seq Read Counts. *Genome Biol* **2014**, *15* (2), R29. https://doi.org/10.1186/gb-2014-15-2-r29.

(13) Zhang, X.; Smits, A. H.; van Tilburg, G. B.; Ovaa, H.; Huber, W.; Vermeulen, M. Proteome-Wide Identification of Ubiquitin Interactions Using UbIA-MS. *Nat Protoc* **2018**, *13* (3), 530–550. https://doi.org/10.1038/nprot.2017.147.

(14) Yu, S.-H.; Ferretti, D.; Schessner, J. P.; Rudolph, J. D.; Borner, G. H. H.; Cox, J. Expanding the Perseus Software for Omics Data Analysis With Custom Plugins. *Curr Protoc Bioinformatics* **2020**, *71* (1), e105. https://doi.org/10.1002/cpbi.105.

(15) 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.

(16) Nicolas, S.; Dohm-Hansen, S.; Lavelle, A.; Bastiaanssen, T. F. S.; English, J. A.; Cryan, J. F.; Nolan, Y. M. Exercise Mitigates a Gut Microbiota-Mediated Reduction in Adult Hippocampal Neurogenesis and Associated Behaviours in Rats. *Transl Psychiatry* **2024**, *14* (1), 1–13. https://doi.org/10.1038/s41398-024-02904-0.

(17) Grabrucker, S.; Marizzoni, M.; Silajdžić, E.; Lopizzo, N.; Mombelli, E.; Nicolas, S.; Dohm-Hansen, S.; Scassellati, C.; Moretti, D. V.; Rosa, M.; Hoffmann, K.; Cryan, J. F.; O'Leary, O. F.; English, J. A.; Lavelle, A.; O'Neill, C.; Thuret, S.; Cattaneo, A.; Nolan, Y. M. Microbiota from Alzheimer's Patients Induce Deficits in Cognition and Hippocampal Neurogenesis. *Brain* **2023**, *146* (12), 4916–4934. https://doi.org/10.1093/brain/awad303.

(18) Gentleman, R. C.; Carey, V. J.; Bates, D. M.; Bolstad, B.; Dettling, M.; Dudoit, S.; Ellis, B.; Gautier, L.; Ge, Y.; Gentry, J.; Hornik, K.; Hothorn, T.; Huber, W.; Iacus, S.; Irizarry, R.; Leisch, F.; Li, C.; Maechler, M.; Rossini, A. J.; Sawitzki, G.; Smith, C.; Smyth, G.; Tierney, L.; Yang, J. Y.; Zhang, J. Bioconductor: Open Software Development for Computational Biology and Bioinformatics. *Genome Biology* **2004**, *5* (10), R80. https://doi.org/10.1186/gb-2004-5-10-r80.

(19) Karpievitch, Y. V.; Dabney, A. R.; Smith, R. D. Normalization and Missing Value Imputation for Label-Free LC-MS Analysis. *BMC Bioinformatics* **2012**, *13 Suppl 16* (Suppl 16), S5. https://doi.org/10.1186/1471-2105-13-S16-S5.

(20) Wei, R.; Wang, J.; Su, M.; Jia, E.; Chen, S.; Chen, T.; Ni, Y. Missing Value Imputation Approach for Mass Spectrometry-Based Metabolomics Data. *Sci Rep* **2018**, *8* (1), 663. https://doi.org/10.1038/s41598-017-19120-0.

(21) Barkovits, K.; Pacharra, S.; Pfeiffer, K.; Steinbach, S.; Eisenacher, M.; Marcus, K.; Uszkoreit, J. Reproducibility, Specificity and Accuracy of Relative Quantification Using Spectral Library-Based Data-Independent Acquisition. *Mol Cell Proteomics* **2020**, *19* (1), 181–197. https://doi.org/10.1074/mcp.RA119.001714.

(22) 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.

(23) Law, C. W.; Alhamdoosh, M.; Su, S.; Dong, X.; Tian, L.; Smyth, G. K.; Ritchie, M. E. RNA-Seq Analysis Is Easy as 1-2-3 with Limma, Glimma and edgeR. *F1000Res* **2016**, *5*, ISCB Comm J-1408. https://doi.org/10.12688/f1000research.9005.3.

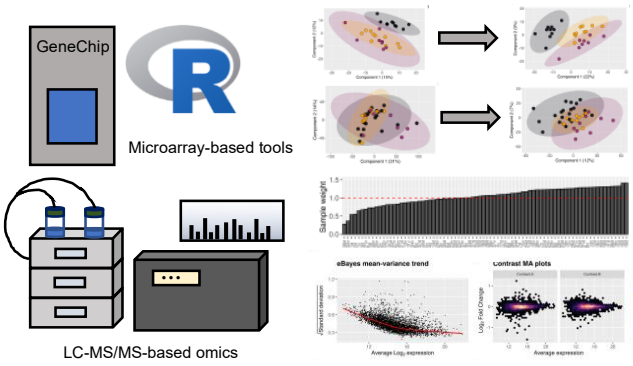
(24) Mahieu, N. G.; Patti, G. J. Systems-Level Annotation of a Metabolomics Data Set Reduces 25 000 Features to Fewer than 1000 Unique Metabolites. *Anal. Chem.* **2017**, *89* (19), 10397–10406. https://doi.org/10.1021/acs.analchem.7b02380.

(25) Gatto, L.; Lilley, K. S. MSnbase-an R/Bioconductor Package for Isobaric Tagged Mass Spectrometry Data Visualization, Processing and Quantitation. *Bioinformatics* **2012**, *28* (2), 288–289. https://doi.org/10.1093/bioinformatics/btr645.
(26) Lazar, C.; Burger, T.; Wieczorek, S. imputeLCMD: A Collection of Methods for Left-Censored Missing Data Imputation, 2022. https://cran.r-project.org/web/packages/imputeLCMD/index.html (accessed 2025-01-28).
(27) Gardner, M. L.; Freitas, M. A. Multiple Imputation Approaches Applied to the Missing Value Problem in Bottom-Up Proteomics. *International Journal of Molecular Sciences* **2021**, *22* (17), 9650. https://doi.org/10.3390/ijms22179650.
(28) 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.
(29) Rohart, F.; Gautier, B.; Singh, A.; Lê Cao, K.-A. mixOmics: An R Package for 'omics Feature Selection and Multiple Data Integration. *PLoS Comput Biol* **2017**, *13* (11), e1005752. https://doi.org/10.1371/journal.pcbi.1005752.
(30) Cox, J.; Hein, M. Y.; Luber, C. A.; Paron, I.; Nagaraj, N.; Mann, M. Accurate Proteome-Wide Label-Free Quantification by Delayed Normalization and Maximal Peptide Ratio Extraction, Termed MaxLFQ. *Mol Cell Proteomics* **2014**, *13* (9), 2513–2526. https://doi.org/10.1074/mcp.M113.031591.
(31) 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.
(32) Kong, A. T.; Leprevost, F. V.; Avtonomov, D. M.; Mellacheruvu, D.; Nesvizhskii, A. I. MSFragger: Ultrafast and Comprehensive Peptide Identification in Mass Spectrometry–Based Proteomics. *Nat Methods* **2017**, *14* (5), 513–520. https://doi.org/10.1038/nmeth.4256.
(33) 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.
(34) Pang, Z.; Chong, J.; Zhou, G.; de Lima Morais, D. A.; Chang, L.; Barrette, M.; Gauthier, C.; Jacques, P.-É.; Li, S.; Xia, J. MetaboAnalyst 5.0: Narrowing the Gap between Raw Spectra and Functional Insights. *Nucleic Acids Res* **2021**, *49* (W1), W388–W396. https://doi.org/10.1093/nar/gkab382.
(35) Li, B.; Tang, J.; Yang, Q.; Li, S.; Cui, X.; Li, Y.; Chen, Y.; Xue, W.; Li, X.; Zhu, F. NOREVA: Normalization and Evaluation of MS-Based Metabolomics Data. *Nucleic Acids Res* **2017**, *45* (W1), W162–W170. https://doi.org/10.1093/nar/gkx449.
(36) Chawade, A.; Alexandersson, E.; Levander, F. Normalyzer: A Tool for Rapid Evaluation of Normalization Methods for Omics Data Sets. *J Proteome Res* **2014**, *13* (6), 3114–3120. https://doi.org/10.1021/pr401264n.
(37) Li, B.; Tang, J.; Yang, Q.; Cui, X.; Li, S.; Chen, S.; Cao, Q.; Xue, W.; Chen, N.; Zhu, F. Performance Evaluation and Online Realization of Data-Driven Normalization Methods

Used in LC/MS Based Untargeted Metabolomics Analysis. *Sci Rep* **2016**, *6*, 38881. https://doi.org/10.1038/srep38881.
(38) De Livera, A. M.; Sysi-Aho, M.; Jacob, L.; Gagnon-Bartsch, J. A.; Castillo, S.; Simpson, J. A.; Speed, T. P. Statistical Methods for Handling Unwanted Variation in Metabolomics Data. *Anal Chem* **2015**, *87* (7), 3606–3615. https://doi.org/10.1021/ac502439y.
(39) Välikangas, T.; Suomi, T.; Elo, L. L. A Systematic Evaluation of Normalization Methods in Quantitative Label-Free Proteomics. *Brief Bioinform* **2016**, *19* (1), 1–11. https://doi.org/10.1093/bib/bbw095.
(40) Jauhiainen, A.; Madhu, B.; Narita, M.; Narita, M.; Griffiths, J.; Tavaré, S. Normalization of Metabolomics Data with Applications to Correlation Maps. *Bioinformatics* **2014**, *30* (15), 2155–2161. https://doi.org/10.1093/bioinformatics/btu175.
(41) Workman, C.; Jensen, L. J.; Jarmer, H.; Berka, R.; Gautier, L.; Nielser, H. B.; Saxild, H.-H.; Nielsen, C.; Brunak, S.; Knudsen, S. A New Non-Linear Normalization Method for Reducing Variability in DNA Microarray Experiments. *Genome Biol* **2002**, *3* (9), research0048. https://doi.org/10.1186/gb-2002-3-9-research0048.
(42) Bolstad, B. M.; Irizarry, R. A.; Astrand, M.; Speed, T. P. A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Variance and Bias. *Bioinformatics* **2003**, *19* (2), 185–193. https://doi.org/10.1093/bioinformatics/19.2.185.
(43) Huber, W.; von Heydebreck, A.; Sültmann, H.; Poustka, A.; Vingron, M. Variance Stabilization Applied to Microarray Data Calibration and to the Quantification of Differential Expression. *Bioinformatics* **2002**, *18 Suppl 1*, S96-104. https://doi.org/10.1093/bioinformatics/18.suppl_1.s96.
(44) Phipson, B.; Lee, S.; Majewski, I. J.; Alexander, W. S.; Smyth, G. K. ROBUST HYPERPARAMETER ESTIMATION PROTECTS AGAINST HYPERVARIABLE GENES AND IMPROVES POWER TO DETECT DIFFERENTIAL EXPRESSION. *Ann Appl Stat* **2016**, *10* (2), 946–963. https://doi.org/10.1214/16-AOAS920.
(45) Sakthivel, K.; Lal, S. B.; Srivastava, S.; Chaturvedi, K. K.; Khan, Y. J.; Mishra, D. C.; Madival, S. D.; Vaidhyanathan, R.; Jha, G. K. A Statistical Approach for Identifying the Best Combination of Normalization and Imputation Methods for Label-Free Proteomics Expression Data. *J Proteome Res* **2025**, *24* (1), 158–170. https://doi.org/10.1021/acs.jproteome.4c00552.
(46) Rocke, D. M.; Durbin, B. A Model for Measurement Error for Gene Expression Arrays. *J Comput Biol* **2001**, *8* (6), 557–569. https://doi.org/10.1089/106652701753307485.
(47) Argelaguet, R.; Velten, B.; Arnol, D.; Dietrich, S.; Zenz, T.; Marioni, J. C.; Buettner, F.; Huber, W.; Stegle, O. Multi-Omics Factor Analysis-a Framework for Unsupervised Integration of Multi-Omics Data Sets. *Mol Syst Biol* **2018**, *14* (6), e8124. https://doi.org/10.15252/msb.20178124.
(48) Liu, R.; Holik, A. Z.; Su, S.; Jansz, N.; Chen, K.; Leong, H. S.; Blewitt, M. E.; Asselin-Labat, M.-L.; Smyth, G. K.; Ritchie, M. E. Why Weight? Modelling Sample and Observational Level Variability Improves Power in RNA-Seq Analyses. *Nucleic Acids Res* **2015**, *43* (15), e97. https://doi.org/10.1093/nar/gkv412.
(49) Ritchie, M. E.; Diyagama, D.; Neilson, J.; van Laar, R.; Dobrovic, A.; Holloway, A.; Smyth, G. K. Empirical Array Quality Weights in the Analysis of Microarray Data. *BMC Bioinformatics* **2006**, *7*, 261. https://doi.org/10.1186/1471-2105-7-261.

(50) Leek, J. T.; Storey, J. D. Capturing Heterogeneity in Gene Expression Studies by Surrogate Variable Analysis. *PLoS Genet* **2007**, *3* (9), 1724–1735. https://doi.org/10.1371/journal.pgen.0030161.
(51) Leek, J. T.; Scharpf, R. B.; Bravo, H. C.; Simcha, D.; Langmead, B.; Johnson, W. E.; Geman, D.; Baggerly, K.; Irizarry, R. A. Tackling the Widespread and Critical Impact of Batch Effects in High-Throughput Data. *Nat Rev Genet* **2010**, *11* (10), 733–739. https://doi.org/10.1038/nrg2825.
(52) Leek, J. T.; Johnson, W. E.; Parker, H. S.; Jaffe, A. E.; Storey, J. D. The Sva Package for Removing Batch Effects and Other Unwanted Variation in High-Throughput Experiments. *Bioinformatics* **2012**, *28* (6), 882–883. https://doi.org/10.1093/bioinformatics/bts034.
(53) Phua, S.-X.; Lim, K.-P.; Goh, W. W.-B. Perspectives for Better Batch Effect Correction in Mass-Spectrometry-Based Proteomics. *Computational and Structural Biotechnology Journal* **2022**, *20*, 4369–4375. https://doi.org/10.1016/j.csbj.2022.08.022.

For TOC only


GeneChip
Microarray-based tools
LC-MS/MS-based omics
eBayes mean-variance trend
contrast MA plots