The Current State of Single-Cell Proteomics Data Analysis

Christophe Vanderaa¹ and Laurent Gatto¹,²

¹Computational Biology and Bioinformatics Unit (CBIO), de Duve Institute, Université catholique de Louvain, Belgium
²Corresponding author: laurent.gatto@uclouvain.be

Published in the Molecular Biology section

Sound data analysis is essential to retrieve meaningful biological information from single-cell proteomics experiments. This analysis is carried out by computational methods that are assembled into workflows, and their implementations influence the conclusions that can be drawn from the data. In this work, we explore and compare the computational workflows that have been used over the last four years and identify a profound lack of consensus on how to analyze single-cell proteomics data. We highlight the need for benchmarking of computational workflows and standardization of computational tools and data, as well as carefully designed experiments. Finally, we cover the current standardization efforts that aim to fill the gap, list the remaining missing pieces, and conclude with lessons learned from the replication of published single-cell proteomics analyses. © 2023 Wiley Periodicals LLC.

Keywords: data analysis • mass spectrometry • proteomics • reproducible research • single-cell

INTRODUCTION

Conducting a principled data analysis is not trivial, especially when technologies and the data they generate increase in complexity at a fast pace. This is particularly true for mass spectrometry (MS)–based single-cell proteomics (SCP) data analysis. Several hurdles need to be overcome to extract biologically meaningful information from these complex data (Vanderaa & Gatto, 2021). Numerous methods exist to correct for technical issues, and each method has its respective advantages and drawbacks. In this review article, we show that the variety of available methods to process proteomics data and the current lack of computational standards has led to great heterogeneity in SCP data analysis practices. This computational heterogeneity is a reflection of the technical heterogeneity, since MS-based SCP has undergone many improvements. For instance, two sample preparation strategies currently co-exist: SCP by label-free quantification (LFQ) and multiplexed SCP (Ctortecka & Mechtler, 2021; Kelly, 2020; Slavov, 2021d). Multiplexing strategies include isobaric labeling, using tandem mass tags (TMT) or non-isobaric labeling, and using mass differential tags for relative and absolute quantification (mTRAQ) (Derks et al., 2022; Petelski et al., 2021). Several chips have been developed starting with the nanoPOTS chip (Zhu et al., 2018b), followed by the N2chip (Woo et al., 2021), the proteoCHIP (Hartlmayr et al., 2021), and the microfluidic SciProChip (Gebreyesus et al., 2022). Efforts have also focused on the automation of sample processing and the successful integration of robot handlers have been reported such as the Mantis (Petelski et al., 2021), the OT-2 (Liang et al., 2021) or the CellenOne (Hartlmayr et al., 2021; Leduc, Huffman, & Slavov, 2022;
Quantitative data processing lacks consensus

Proteomics data analysis encompasses three main tasks: spectral data processing, quantitative data processing, and downstream data analysis. Spectral data processing identifies and quantifies the peptides from the acquired MS spectra. Assigning peptide sequences to the MS spectrum was spotlighted as an important challenge for SCP data analysis (Slavov, 2021a), and several groups have contributed to methodological and software improvements. For instance, Yu et al. extended the match between run (MBR) algorithm from MaxQuant to TMT data, taking advantage of the quantification data present in unidentified MS2 spectra (Yu, Kyriakidou, & Cox, 2020). The iceR package also propagates information across runs. The algorithm dramatically improves peptide identification and outperforms MBR (Kalxdorf, Miller, Stegle, & Krijgsved, 2021). Unfortunately, iceR is only applicable to label-free data. Another approach to improve peptide identification is to increase the confidence of matching by re-scoring. Re-scoring uses the annotations generated by the search engines such as the deviation between expected and measured elution times or m/z, the peptide length, or the ion charge (Van Der Watt et al., 2021) to update the score or probability that measured spectra correctly match spectra from a theoretical or empirical spectral library. DART-ID, a Bayesian framework to update posterior error probabilities based on an accurate estimation of elution times, has been applied to SCP data and showed a significant increase in the number of identified spectra (Chen, Franks, & Slavov, 2019). Others have also improved the Percolator re-scoring algorithm for SCP experiments (Fondrie & Noble, 2020; Fondrie & Noble, 2021), although the measured improvements were subtle. While these developments considerably improve the quality of spectrum identification, no dedicated developments in quantitative data processing have been reported.

Quantitative data processing plays a critical role in overcoming many technical artifacts and satisfying downstream analysis requirements. It consists of several steps. Quality controls ensure that the analyzed data are composed of reliable information and remove features of low quality that could otherwise compromise the validity of the results. Aggregation combines peptide level data into protein level data. Log-transformation shapes the data so that the quantitative values follow...
normal distributions. Imputation generates estimates for missing values. Finally, normalization and batch correction aim to remove technical differences between samples and are essential to avoid biased results. Each of these steps is implemented using different methods. For instance, many methods exist for missing value imputation: replace by zero, replace with random values sampled from an estimated background distribution, replace by values estimated from the K-nearest neighbors (KNN), etc. The imputation methods have different underlying assumptions that have been extensively reviewed in the bulk proteomics field (Bramer, Irvahn, Piehowski, Rodland, & Webb-Robertson, 2021), but further research is required to assess whether these assumptions remain valid for SCP data. In addition to choosing the right method, finding a correct sequence of steps is another challenge. For instance, batch effects influence missing data, and vice versa (Vanderaa & Gatto, 2021). It has been suggested to correct for batch effects before imputation (Cuklina et al., 2021), but batch correction methods such as ComBat (Johnson, Li, & Rabinovic, 2007) break with highly missing data, as in SCP data.

As of today, developing computational workflows for SCP quantitative data processing requires expert knowledge. We refer to “computational workflow” or “computational pipeline” as the sequence of steps and methods that process quantification data for downstream statistical testing or visualization. Computational workflows are built from scratch and their development often lacks an explicit rationale. Since we lack systematic comparisons, benchmarks, or guidelines, the processing approaches become fundamentally different between publications. To illustrate our claim, we review the computational approaches from several studies that have shaped the SCP landscape since 2018 (Table 1). These studies present significant contributions to the field and showcase applications on actual single cells as opposed to bulk lysate dilutions. Five studies supplemented their publication with material allowing to repeat, at least partially, their computational analysis. Three studies from the Slavov Lab provide the R code and the data required to fully repeat their results (Derks et al., 2022; Leduc et al., 2022; Specht et al., 2021). However, the code is poorly documented and difficult to re-use by other labs. Schoof et al. also offer the data used to repeat their study and distribute their computational workflow as a documented Python library, sceptre (Schoof et al., 2021). Their library heavily relies on scanpy, a popular Python library for scRNA-seq analysis (Wolf, Angerer, & Theis, 2018). Finally, Brunner et al. provide a Python script that also relies on scanpy, but it lacks an explicit link with the input data (Brunner et al., 2022). Based on the available material, scripts for (Brunner et al., 2022; Derks et al., 2022; Leduc et al., 2022; Schoof et al., 2021; Specht et al., 2021), or the methods section for the others, we constructed Figure 1. We divide the workflow steps into eight general categories and further group the different steps depending on whether they are applied at the precursor/peptide-to-spectrum match (PSM) level, peptide level, protein level, or implicitly embedded in an MS data preprocessing software.

Several conclusions can be drawn from Figure 1. First, one publication corresponds to one workflow. This variability cannot be explained solely by different experimental protocols. The computational pipelines by Schoof et al. and Specht et al. differ substantially, while their TMT-based acquisition protocols are closely related (Schoof et al., 2021; Specht et al., 2021), and the computational pipeline by Liang et al. for processing LFQ data (Liang et al., 2021) is more similar to the TMT processing workflow of Williams et al. than its LFQ alternative. Moreover, some publications provide a minimalist computational workflow with only 3 steps while others perform extensive processing with 20 steps. These observations highlight the lack of consensus and the need to identify critical steps in computational pipelines. Second, some processing steps are applied at the peptide level while others are applied at the protein level. For instance, Budnik et al. perform normalization at the peptide level, whereas Dou et al. perform normalization at protein level (Fig. 1I). A clear pattern is that most pipelines process the data at the protein level, which is questionable, since processing data at an earlier stage could avoid the propagation of technical artifacts to the protein data (Lazar, Gatto, Ferro, Bruley, & Burger, 2016; Sticker, Goeminne, Martens, & Clement, 2020). Third, a great majority of the methods are taken from bulk proteomics. We foresee that developing new methods that account for the properties inherent to single-cell data would significantly improve the workflows. For instance, batch correction could benefit from dedicated single-cell methods, as the strong
Figure 1 Overview of quantitative data processing workflows. (A-H) The workflows are split into eight categories represented by different colors. Each category contains a set of methods that are represented by different shapes (see Table 2 for a description of the methods). Each point indicates which method (column) is implemented in which publication (row). Some methods are used in several workflows (points align vertically) and some workflows used several methods (points align horizontally). (I) Summary of the sequence of processing steps for each workflow. Depending on the workflow, the processing steps are implicitly executed by the MS data software and applied at the PSM/precursor, peptide or protein level. Colors and shapes follow the structure from the previous panels. Each horizontal line represents a workflow and should be read from left to right.
| Study                          | Publication date | Raw data analysis       | Labeling   | Script | Throughput |
|-------------------------------|------------------|-------------------------|------------|--------|------------|
| Zhu et al. (2018a)            | Sep 2018         | MaxQuant/Andromeda      | —          | —      | 6          |
| Budnik, Levy, Harmange, and Slavov (2018) | Oct 2018         | MaxQuant/Andromeda      | TMT-10     | —      | 190        |
| Dou et al. (2019)             | Oct 2019         | MS-GF+, MASIC           | TMT-10     | —      | 72         |
| Zhu et al. (2019)             | Nov 2019         | MaxQuant/Andromeda      | —          | —      | 28         |
| Cong et al. (2020)            | Jan 2020         | MaxQuant/Andromeda      | —          | —      | 4          |
| Tsai et al. (2020)            | May 2020         | MaxQuant/Andromeda      | TMT-11     | —      | 104        |
| Williams et al. (2020)        | Aug 2020         | MaxQuant/Andromeda      | —          | —      | 17         |
| Williams et al. (2020)        | Aug 2020         | MaxQuant/Andromeda      | TMT-11     | —      | 152        |
| Liang et al. (2021)           | Dec 2020         | FragPipe/MSFragger      | —          | —      | 3          |
| Specht et al. (2021)          | Jan 2021         | MaxQuant/Andromeda      | TMT-11,TMT-16 | R  | 1490       |
| Cong et al. (2021)            | Feb 2021         | PD/SEQUEST              | —          | —      | 6          |
| Schoof et al. (2021)          | Jun 2021         | PD/SEQUEST              | TMT-16     | Python | 2025       |
| Woo et al. (2021)             | Oct 2021         | MaxQuant/Andromeda      | TMT-16     | —      | 108        |
| Brunner et al. (2022)         | Feb 2022         | DIA-NN                  | —          | Python | 231        |
| Leduc et al. (2022)           | Mar 2022         | MaxQuant/Andromeda      | TMT-18     | R      | 1556       |
| Woo et al. (2022)             | Mar 2022         | MaxQuant/Andromeda      | —          | —      | 155        |
| Webber et al. (2022)          | Apr 2022         | PD/SEQUEST              | —          | —      | 28         |
| Derks et al. (2022)           | Jul 2022         | DIA-NN                  | mTRAQ-3    | R      | 155        |

These studies were published between 2018 and 2022. MaxQuant, FragPipe, Proteome Discoverer (PD), and DIA-NN are software tools to conduct peptide identification and quantification. The peptide identification is performed by underlying search engines such as Andromeda, MS-GF+, MSFragger or SEQUEST. Multiplexing relies on TMT or mTRAQ labeling while no labeling implies an LFQ approach. Some publications link to associated computational scripts to reproduce the analysis that were written either in Python or R. The throughput is expressed in number of cells retained after sample quality control, if any (Fig. 1A).
Table 2 Description of SCP Data Processing Methods

| Step                      | Method                        | Description                                                                                                                                 |
|---------------------------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| Sample quality control    | Failed runs                   | Remove samples that belong to an MS acquisition that failed based on the number of PSMs.                                                      |
|                           | Missing data                  | Remove samples that contain too many missing values. Too many is defined by the authors either using the percentage missing value per cell or the number of identified proteins per cell. |
|                           | Median CV                     | Remove samples for which the median coefficient (CV) of variation is higher than a user provided threshold. CVs are computed from peptides belonging to the same proteins and CVs per cell across proteins are summarized using the median. |
|                           | MAD sum intensity             | Remove samples for which the total protein signal intensity per cell falls outside the median absolute deviation (MAD). Briefly, the summed intensities are computed for each cell. The median of the absolute differences between the summed intensities and the computed median is the MAD. All cells that have their summed intensity higher than the median plus MAD, or lower than the median minus MAD, are removed. |
|                           | Grubb’s test                  | Iteratively remove outlying cells based on the ratio of the absolute deviation to the mean of all cells and the standard deviation. Outlying cells are removed based on a significance threshold ($\alpha = 5$). |
| Feature quality control   | Contaminants and decoys       | Remove identified peptides that match the contaminant and/or the decoy database.                                                              |
|                           | Empty signal                  | Remove features for which no signal is recorded across all single cells.                                                                      |
|                           | FDR filtering                 | Remove peptides or proteins that have an associated false discovery rate (FDR) greater than a user provided threshold, usually 1%. FDR is computed by the search engine during peptide identification. |
|                           | Missing data                  | Remove peptides or proteins that contain too many missing values. Too many is defined by the authors either using the percentage missing value per feature or the number of cells the feature was identified in. |
|                           | Unique peptides per protein   | Remove proteins that were identified and quantified from less than 2 unique peptides.                                                        |
|                           | Spectral purity               | Remove PSMs that have a low spectral purity, that is the proportion of signal from the identified peptide compared to signal from background or co-eluting peptide. Spectral purity is computed by the search engine during peptide identification. |
|                           | Sample to carrier ratio       | Remove the PSMs for which the ratio between the average intensity across single cells over the signal in the carrier channel is higher than a user provided threshold, usually 10%. |

(Continued)
| Step          | Method | Description                                                                                                                                 |
|--------------|--------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Imputation   | KNN    | Missing values are imputed using k-nearest neighbors (KNN) averaging. In other words, for each cell, the missing value for a protein is replaced |
|              |        | with the average intensity of K most similar cells (in Euclidean space) for which that protein is not missing. K is defined by the user or |
|              |        | optimized by maximizing the silhouette width between known cell types.                                                                     |
|              | Zero   | Missing values are replaced by a zero value.                                                                                                                                                      |
|              | Normal | Missing values are replaced by values sampled from a normal distribution. The mean of the distribution is defined as the global mean of |
|              |        | non-missing values downshifted by a user provided value; the standard deviation of the distribution is set by the user as well.           |
| Log-transform| log2   | Replace quantitative values by the $\log_2$ of the values.                                                                                                                                         |
|              | log 1p | Replace quantitative values by the $\ln$ of the values with a pseudocount of one.                                                                                                                  |
| Aggregation  | Random | Pick one feature to represent the aggregated feature. This method was used to aggregate PSMs to charged peptides. In this context there are |
|              | selection | about 0.1% of charged peptides that originate from more than one PSM.                                                                         |
|              | Summed | Combine quantitative data from multiple features by summing their intensities.                                                                                                                       |
|              | Top N  | Combine quantitative data from multiple features by summing intensities of the N most intense features.                                                                                              |
|              | Median | Combine quantitative data from multiple features by taking the median of their intensities.                                                                                                        |
|              | iBAQ   | Intensity based absolute quantification (iBAQ) combines peptides into protein quantifications by summing the peptide intensities divided by |
|              |        | the number of theoretically observable peptides.                                                                                                                                                |
|              | maxLFQ | Protein quantifications are computed from peptide quantification using delayed normalization and maximal peptide ratio extraction. See (Cox |
|              |        | et al., 2014) for detailed information on the method.                                                                                                                                              |
| Normalization| Sample | Quantitative values for each single cell are divided or subtracted by the median for that cell.                                                                                                     |
|              | median |                                                                                                                                                                                            |
|              | Feature | Quantitative values for each feature are divided or subtracted by the mean for that feature.                                                                                                       |
|              | median |                                                                                                                                                                                            |
|              | Total  | Quantitative values for each feature are divided or subtracted by the median for that feature.                                                                                                     |

(Continued)
| Step | Method                  | Description                                                                                                                                                                                                 |
|------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|      | Quantile                | Quantitative values are transformed so that the quantile distribution is aligned between all single cells.                                                                                                     |
|      | Standardization         | Quantitative values for each feature are subtracted by the average and divided by the standard deviation.                                                                                                      |
| Batch correction | Reference               | For all cells within a batch and for each feature, the quantitative data are divided or subtracted by the reference samples within that run.                                                                |
|      | Median                  | For all cells within a batch and for each feature, the quantitative data are divided or subtracted by the median values within that run.                                                                    |
|      | ComBat                  | Batch effects are removed using linear regression with an Empirical Bayes framework. The method models the quantifications of each protein with respect to a batch factor as well as the biological factors to protect from correction. |
|      | Limma                   | This method is very similar to ComBat. It allows for more than one batch factors.                                                                                                                                 |
| Other | Isotope impurity correction | The signal associated with a label is corrected by removing the signal associated with impurities from the other labels. This is performed using a compensation matrix provided by the label manufacturer or from theoretical isotypic composition. This is only applicable for multiplexed experiments. |
|      | Quantification thresholding | Any quantitative value below a user provided threshold is set to be missing. See (Cheung et al., 2020) for more detailed information on how to set the threshold.                                                   |
|      | S/N computation         | Convert MS signal intensities to signal-to-noise ratio (S/N). The noise signal is estimated by the spectrum quantification software.                                                                          |
The dependency between batch effects and missing data requires robust and tailored models (Vanderaa & Gatto, 2021). Horizontal integration of samples from different batches is an active field of research in single-cell omics (Argelaguet, Cuomo, Stegle, & Marioni, 2021) that will probably be beneficial to the SCP community.

Figure 1 highlights the need for a better understanding on how to process and model SCP data. Identifying which workflows perform best, or demonstrating whether a new workflow improves performance, requires thorough computational benchmarking. The scRNA-seq field already offers tools for method benchmarking that could readily be used for SCP applications (Germain, Sonrel, & Robinson, 2020; Tian et al., 2019). In order to run these tools, the computational workflows should be accessible to the benchmarking software. Another key consideration is that benchmarking datasets are required to enable an objective comparison between computational pipelines. While many SCP datasets are available from public sources, these data are provided in different formats. Proper benchmarking necessitates a standardization of the computational pipelines and the data. In the next section, we cover the recent developments that attempt to harmonize quantitative data processing for SCP.

CURRENT SOLUTIONS FOR QUANTITATIVE DATA PROCESSING
We recently published an R/Bioconductor package called scp (Table 3). First, scp is thoroughly documented, as we want to facilitate its re-use. Second, it is designed as a modular tool where each processing step, such as those defined in Figure 1, can easily be chained and return a consistent and standardized output format. Third, the software is part of the Bioconductor project (Huber et al., 2015), which is well known for exemplary coding practices and promotes long term maintenance. Fourth, scp can be integrated with other tools that rely on QFeatures and SingleCellExperiment, two data structures widely used for proteomics and single-cell data analysis, respectively (Amezquita et al., 2020; Gatto & Vanderaa, 2022). Finally, scp is maintained and improved to include current state-of-the-art methods. For instance, it re-implements functionality from the SCoPE2 script released by Specht et al. (2021). Next to scp, Schoof et al. developed sceptre, a Python module that implements their computational workflow (Schoof et al., 2021) (Table 3). The code is well documented, modular, and relies on scannpy (Wolf et al., 2018), a Python data structure equivalent to SingleCellExperiment. The tool, however, lacks flexibility as it was developed primarily to offer a reproducible data analysis environment. Minor code refactoring could overcome this lack of flexibility.

Computational solutions require data to develop, test, and benchmark individual methods and complete workflows. We therefore also recently developed another R/Bioconductor package, scpdata, that distributes curated SCP datasets ready for analysis (https://bioconductor.org/packages/release/data/experiment/html/scpdata.html; Table 3). The datasets were retrieved from published work and are accessible using a single command. The standardization effort provides a thoroughly annotated and consistent data structure, facilitating data analysis with tools such as scp. Furthermore, scpdata relies on Bioconductor’s storage services, ExperimentHub (Morgan & Shepherd, 2022), that offers cloud-based data access. Easy access and consistent formats enable method development on a variety of different datasets, avoiding dataset-specific over-fitting. The scpdata implementation can also be used for benchmarking, although ground truths are missing to perform accuracy validation.

Standardized data processing tools allow going beyond the reproduction of existing SCP data analyses; they enable their replication. While reproduction allows others to re-generate the same results using the same software or computational setup, replication uses different software or analysis methods to generate the same, or equivalent, results. Replication, therefore, consolidates our trust in previous work. Although a replicable analysis does not imply the results are correct, it guarantees the results do not rely on undocumented steps or on software peculiarities. For instance, we have shown that the SCoPE2 analysis script by Specht et al. could be fully replicated using scp and scpdata (Specht et al., 2021). Further, replication efforts have additional benefits. Replication can highlight hurdles that prevent accurate data analysis. Continuing with the SCoPE2 example, our replication study identified batch effects and missing data, and their dependence, as prominent challenges that future SCP computational tools will need to tackle. Another beneficial side effect is that replication studies are easily
| Tool            | Language | Description and link                                                                 |
|-----------------|----------|--------------------------------------------------------------------------------------|
| Data            | scpdata  | R/Bioconductor package providing published SCP datasets ready for analysis in R.     |
|                 |          | [https://bioconductor.org/packages/release/data/experiment/html/scpdata.html](https://bioconductor.org/packages/release/data/experiment/html/scpdata.html) |
| Processing      | scp      | R/Bioconductor package to process quantitative SCP data, supporting all steps and methods depicted in Figure 1. |
|                 |          | [https://bioconductor.org/packages/release/bioc/html/scp.html](https://bioconductor.org/packages/release/bioc/html/scp.html) |
|                 | sceptre  | Python package extending the functionalities of scanpy to analyse multiplexed SCP data. It implements the Schoof et al., 2021 workflow in Figure 1. |
|                 |          | [https://github.com/bfurtwa/SCeptre](https://github.com/bfurtwa/SCeptre) |
| Reproduction    | SCP.replication | Website with 8 SCP data analysis articles that replicate published studies.         |
|                 |          | [https://uclouvain-cbio.github.io/SCP.replication](https://uclouvain-cbio.github.io/SCP.replication) |
|                 | SCeptre notebooks | Code repository containing a set of jupyter notebooks that reproduce the results presented in Schoof et al., 2021. |
|                 |          | [https://github.com/bfurtwa/SCeptre/tree/master/Schoof_et_al/code](https://github.com/bfurtwa/SCeptre/tree/master/Schoof_et_al/code) |
|                 | SlavovLab code | Set of code repositories that reproduce the results published by the Slavov lab. |
|                 |          | [https://github.com/Single-cell-proteomics](https://github.com/Single-cell-proteomics) |
|                 | Brunner code | Code repository with a jupyter notebook to reproduce the results in Brunner et al., 2022. |
|                 |          | [https://github.com/theislab/singlecell_proteomics](https://github.com/theislab/singlecell_proteomics) |
| Tutorials       | SCP workshop | Online tutorial with hands-on exercises.                                           |
|                 |          | [https://lgatto.github.io/QFeaturesScpWorkshop2021](https://lgatto.github.io/QFeaturesScpWorkshop2021) |
Figure 2  Impact of quantitative data processing workflows on the Schoof et al., 2021 dataset. All results presented in this figure were computed from the protein data processed by the corresponding workflow. (A) The silhouette widths provide a measure of cell type consistency. Cell types are defined based on the known labels provided with the data. The silhouettes were computed using the Jaccard similarity on the shared nearest neighbor graph (K = 15). (B) Pearson correlations are computed between all cells with the same cell type label and provide a measure of protein quantification consistency. (C) Unsupervised clustering is performed using Louvain clustering (Blondel, Guillaume, Lambiotte, & Lefebvre, 2008) on a shared nearest neighbor graph (K = 15). The heatmap illustrates the cell distributions across the clustering results computed from the SCoPE2 and the SCeptre workflow. Frequency is given as the number of cells. (D and E) The first 2 principal components of the protein data. Colors indicate the cell type labels.

repurposed for demonstration. As part of this overview, we offer a website, SCP.replication, with replication studies that demonstrate the analysis of SCP data using the scp and scp-data packages (Table 3). It contains several replication articles, spanning TMT and LFQ protocols as well as DDA and DIA data. We also converted the replication material into openly available workshop material. The workshop can be run without prior installation requirements thanks to the Orchestraplatform supported by the Bioconductor project.

DIFFERENT WORKFLOWS LEAD TO DIFFERENT RESULTS
To illustrate the impact of data processing on the analysis outcome, we compare two computational workflows: SCoPE2, released by Specht et al., and SCeptre1, released by Schoof et al. We retrieved the data from scpdata and recreated both pipelines with scp. SCeptre uses a custom implementation for batch correction that is provided by the sceptre library. Thanks to two R packages, reticulate (Ushey, Allaire, & Tang, 2022) and zellkonverter (Zappia & Lun, 2021), we could easily integrate the Python utilities to scp. We then ran the two workflows on the two datasets and compared the results in Figures 2 and 3. Both the cell type consistency, given by the silhouette widths (Figs. 2A & 3A), and the within cell type correlation distributions (Figs. 2B & 3B) are affected by the computational workflow.

1We use sceptre (all lowercase) to refer to the Python library, and we use SCeptre (uppercase “SC”) to refer to the computational workflow.
Figure 3  Impact of quantitative data processing workflows on the Specht et al., 2021 dataset. All results presented in this figure were computed from the protein data processed by the corresponding workflow. (A) The silhouette widths provide a measure of cell type consistency. Cell types are defined based on the known labels provided with the data. The silhouettes were computed using the Jaccard similarity on the shared nearest neighbor graph ($K = 15$). (B) Pearson correlations are computed between all cells with the same cell type label and provide a measure of protein quantification consistency. (C) Unsupervised clustering is performed using Louvain clustering (Blondel et al., 2008) on a shared nearest neighbor graph ($K = 15$). The heatmap illustrates the cell distributions across the clustering results computed from the SCoPE2 and the SCeptre workflow. Frequency is given as the number of cells. (D & E) The first 2 principal components of the protein data. Colors indicate the cell type labels. workflow. The effect is most visible on the principal component analysis (PCA) plots for the Specht et al. (2021) dataset (Fig. 3D & 3E). The data processed with the SCeptre workflow are organized in a horseshoe shape in lower dimensions. This effect is commonly attributed to the presence of a latent continuous variable or gradient, and a careful data exploration revealed the presence of residual batch effects (Fig. 4D). Unsupervised clustering of the processed protein data leads to different groups, even though identical methods and parameters were used (Figs. 2C & 3C). The number of identified clusters differs between the data processed by SCoPE2 and SCeptre. Furthermore, some clusters from one workflow are scattered throughout clusters from the other workflow. Unsupervised clustering is used to identify groups of cells from which to infer a functional state. So, different clustering results can lead to different biological interpretation. To objectively quantify the performance improvement between the two workflows, we need controlled designs with known expectations. Benchmarking efforts using mixture designs have already been performed for scRNA-seq (Tian et al., 2019). Tian et al. acquired both intact single cells and diluted bulk lysates from three cell lines mixed at different proportions and different quantities. These data were used to assess the ability of computational pipelines to retrieve the original design. However, increased performance of a computational workflow on a single data set is not sufficient. Different types of SCP data exist (LFQ & TMT data, DDA & DIA data, orbitrap & time of flight data, etc.), and computational workflows may not generalize well
Confounding effects cause undesired variability. (A) MS drift for the KLLEGEEESR peptide in the Specht et al., 2021 dataset (Specht et al., 2021). The peptide data was processed using scp according to the script provided along the original paper up to log-transformation (Fig. 1I). The batch index is ordered by time of acquisition. The MS drift is highlighted by a grey line computed using LOESS. (B) PCA plot that replicates the results by Brunner et al. (Brunner et al., 2022). The protein data was processed using scp as an R alternative to the Python script provided along the original paper (Fig. 1I). Shapes represent the cell cycle stage; colors represent the day of acquisition. (C) PCA plot that replicates the results by Williams et al. (Williams et al., 2020). The protein data was processed using scp according to the workflow described in the Methods section of the original paper (Fig. 1I). Shapes represent the cell type; colors represent the TMT label. (D) Same as Figure 3D, but colored according to the MS acquisition batch (n = 149 batches).

for all SCP protocols and data. In other words, we do not expect a single computational workflow to perform optimally for all SCP datasets, as different workflows may focus on different characteristics in the data such as the pattern of missing values. Therefore, strengths and weaknesses of computational pipelines need to be identified and documented. To evaluate this, a community effort could replicate the mixture design using the different SCP protocols, as already seen for scRNA-seq (Mereu et al., 2020). This approach would make it possible to further assess robustness of the workflows on different types of SCP data. Although published SCP datasets are available and, as demonstrated above, they can be processed using standardized software, we are still lacking the data needed to quantify the performance of computational workflows.

LESSONS LEARNED FROM REPLICATION OF SCP DATA ANALYSIS

When building the replication studies, we faced several practical challenges regarding the computational analysis of SCP data. In this section, we provide several recommendations that we hope will help practitioners to improve and facilitate future SCP data analyses. While several of these lessons are applicable to bulk proteomics, we focus on SCP examples here.

Complex Analyses Require Suitable Tools

Several search engines for raw MS data identification have been applied to SCP data: MaxQuant/Andromeda, SEQUEST, MS-GF+, and MSFragger (Table 1). While MaxQuant/Andromeda is by far the most popular among those search engines, several
authors observed that it was the worst performing tool in the context of SCP (Cong et al., 2021; Liang et al., 2021; Tsai et al., 2021; Van Der Watt et al., 2021). These observations indicate that one should compare the results of several search engines in order to maximize the number of reliable spectrum identifications. Moreover, all search engines applied to SCP data so far have been developed for bulk proteomics data. Boekweg et al. showed that SCP data have different spectral properties compared to bulk proteomics data and, therefore, the field would benefit from new search engines developed specifically for SCP data (Boekweg et al., 2022). Quantitative data processing is also carried out using different analysis software. Spreadsheet-based and graphical user interface-driven software are currently predominant. However, method development is facilitated by programming languages such as R and Python. Utilizing programming languages involves a steep learning curve and is often limited to expert data analysts, but it offers access to more advanced methods and is a direct and proven solution for assessable, replicable, and reusable computational analyses. Finally, an important criterion when choosing software is its maintenance activity. It has recently been shown that software accuracy best correlates with the author’s commitment to its maintenance (Gardner et al., 2022).

**Consistent Input Formats Facilitate Data Analysis**

When performing data analysis, formatting input data is a time-consuming and error-prone task. To limit this hurdle, scp and sceptre implement functionality to read structured data tables. The sceptre implementation is designed to read Proteome Discoverer tables and requires plate annotations and FACS data. The scp implementation is more generic and has been used to read tables from MaxQuant, Proteome Discoverer, and DIA-NN, and requires a sample annotation table. Both implementations require consistent inputs, as provided by software, that export consistent output tables. Conversely, sample annotation tables depend on the user. When building the scpdata packages, we realized that annotation tables are often lacking and thus needed to be created from the methods section or from the file names. In other cases, annotations were available through different files and required heavy data wrangling. This process is labor-intensive and error-prone. We suggest creating consistent annotation tables where each row represents a sample (single-cell, TMT carrier, negative control, etc.) and columns represent technical or biological variables (Gatto et al., 2022). These variables are then used during statistical modelling to distinguish biological and technical variability. The annotation tables also require thorough documentation of the information each column contains. Consistent input formats streamline data analysis, facilitate the evaluation of the experimental design (Fig. 4B & 4C), and provide the information needed for principled data modelling.

**Beware of Confounding Effects**

All experiments are prone to technical variability and noise, but good experimental design and principled data analysis can disentangle this undesired technical variability from the desired biological variability. This is also the case for bulk proteomics, but the technical challenges are exacerbated when dealing with single-cell data (Vanderaa & Gatto, 2021). Real-life SCP experiments require the acquisition of over hundreds or thousands of cells spread across many MS runs. Each acquisition run is prone to technical factors that influence the quantification results. For instance, MS signal drift arises from a continuous distortion of the signal between sequential runs, as was already described for bulk proteomics (Cuklina et al., 2021). Figure 4A confirms that MS drift is also present in SCP data. Differences between cells over acquisition time are higher than the differences between cell types at each time point. A careful experimental design has spread the two cell types over time, and hence the biological effects can be decoupled from MS drift, thanks to batch correction or statistical modeling. Neglecting this technical effect can have dramatic consequences. As an example, Figure 4B depicts an SCP experiment where single cells are blocked at one of four different cell division stages. Unfortunately, these four categories were acquired sequentially, confounding the desired biological and unwanted technical sources of variation, as well as impairing deconvolution of the technical and the biological variability. When conducting multiplexed experiments, one must keep in mind that the labels also influence single-cell quantification (Schoof et al., 2021). This effect was overlooked in Figure 4C, where each TMT tag is assigned to only one cell type. Again, this impairs the computational modeling of the TMT effects, although in this case the biological variance is more important than the variance associated with the TMT label.
To overcome such a confounding effect, it is crucial to carefully design an experiment using an adequate statistical blocking scheme and collect data about any technical factors that may influence the results of an experiment such as LC-MS/MS maintenance, the type of instrument used, the multiplexing labels, the sample preparation batch, the lab that performed the experiment, or cell culture batch. Known biological factors should also be gathered such as cell line, subject ID, or treatment condition. Single cells should then be randomized across all the identified factor levels. (Gatto et al., 2022; Petelski et al., 2021; Schoof et al., 2021; Specht et al., 2021). Unfortunately, technical constraints may not allow for randomized designs. For instance, precious samples from patients may need to be processed on-the-fly. Hence, the patient identity and their clinical phenotype will inevitably be correlated with other technical factors. In scRNA-seq, pseudo-bulking has been successfully applied to perform differential expression analysis when the experimental condition is correlated with the subject. Pseudo-bulking consists of aggregating cells belonging to the same individual after identification and separation of the cell (sub-)populations (Crowell et al., 2020; Lun & Marioni, 2017). However, how to aggregate proteomics data is still to be explored. Another alternative is to use linear mixed models. Although these models are computationally more expensive, they overcome the need for aggregation by modelling protein estimates from the peptide data (Goeminne, Sticker, Martens, Gevaert, & Clement, 2020). Finally, dedicated efforts are required to better monitor and control batch effects in SCP data.

Additionally, it is important to validate the batch correction and exclude residual batch effects. For instance, we observed that the cell types in the Specht et al. (2021) dataset cannot be separated in lower dimensions when processed by the SCeptre pipeline. This is because most of the variability is explained by residual batch effects (Fig. 4D) indicating that the batch correction method implemented in SCeptre is not suited for this dataset. We recommend exploring the effect of technical variables in lower dimensions to offer an intuition on residual batch effects. Batch correction assessment can be adapted from bulk proteomics, such as comparing correlations within batches and within conditions or correlations between unrelated peptides and peptides from the same protein. We refer to Cuklina et al. (2021) for a thorough discussion.

Quantitative Data Processing Depends on Downstream Analyses

The purpose of quantitative data processing is to prepare the data for downstream analyses. Downstream analyses process data into interpretable statistical results that in turn can lead to new biological knowledge. Several approaches have successfully been applied to SCP data. Dimension reduction condenses the data into fewer variables. These data embeddings are often used for data visualization, clustering, or trajectory analysis. Differential abundance or differential detection tests identify proteins whose abundance are statistically different, given a distribution model between experimental conditions or cell clusters. For the latter, significant proteins are called marker proteins and can be used to perform cell type identification. Trajectory analysis infers differentiation or response tracks in the data by estimating a pseudo-timeline. It was recently speculated that SCP technologies enable the study of direct protein regulatory interactions, opening the analysis to the discovery of new regulatory mechanisms using an untargeted approach (Slavov, 2022). Quantitative data processing and downstream analysis could also be combined in a single statistical framework. While no such methods have been developed for SCP data yet, there are several examples from the scRNA-seq field that could be adapted to SCP. For instance, scVI (Lopez, Regier, Cole, Jordan, & Yosef, 2018) or ZINB-WaVE (Risso, Perraudeau, Gribkova, Dudoit, & Vert, 2018) implement a modeling procedure that performs normalization, batch correction, imputation, and dimension reduction as part of the same fitting process. Furthermore, scVI offers a Bayesian approach to perform hypothesis testing directly from the estimated model parameters. While these compelling modeling procedures get rid of many processing steps, they still require thorough sample quality control and feature selection. Whatever the chosen downstream method is, quantitative data processing workflows must match the underlying assumptions and data distributions. For instance, dimension reduction using PCA requires data imputation, but dimension reduction using non-iterative partial least squares (NIPALS) offers a similar alternative that is robust against missing values, and thus does not require imputation (Andrecut, 2009; Vanderea & Gatto, 2021). Quantitative values need to be batch-corrected when running a t-test, otherwise the results will become biased and inaccurate. However, when using linear regression, technical factors...
can be directly included as part of the model and do not require previous batch correction (Ritchie et al., 2015).

CONCLUDING REMARKS
While standardized SCP protocols are applied outside the pioneering labs (Leduc et al., 2022; Liang et al., 2021; Petelski et al., 2021; Tsai et al., 2021), computational workflows to process SCP data still lack any form of standardization. The overwhelming diversity of pipelines makes it difficult to make informed decisions as how to analyze SCP data. We provide important guidelines to orient the design of the data analysis. First, SCP designs and data analyses are complex, and analysis tools should be carefully chosen. Second, robust data analysis relies on consistent and standardized data formats. Standardized data structures should facilitate sample annotation, biological, and technical factors that influence data acquisition. Third, accounting for batch effects is essential to avoid assigning biological discoveries to technical variation, especially for SCP experiments, comprising ever-increasing numbers of single cell samples. Finally, the processing of quantitative data highly depends on the research question at hand and, hence, on the downstream analysis to perform. It is not possible to define a good computational workflow without defining the task to accomplish.

More work is required to offer clear answers on how to set up optimal SCP experimental designs and associated computational pipelines. The field still lacks understanding of the impact of each processing step on the final results. Workflows are still built based on empirical and arbitrary decisions. As the technology gains momentum and more groups start to embrace SCP, setting more complex designs, standardized and benchmarked computational pipelines are needed to guarantee sound data interpretation. Indeed, strong data analysis principles and frameworks will enable the technology to reach its full potential. Low quality results generated by flawed analysis practices could penalize the field instead of incentivize for better analysis. The implementation of scp/scpdata and sceptre represent strong foundations that can support computational benchmarking efforts.

ACKNOWLEDGMENTS
This work was funded by a research fellowship of the Fonds de la Recherche Scientifique-FNRS.

AUTHOR CONTRIBUTION
Christophe Vanderaa: Data curation, formal analysis, software, visualization, and writing original draft. Laurent Gatto: Conceptualization, funding acquisition, methodology, project administration, software, supervision, writing original draft, writing review, and editing.

CONFLICT OF INTEREST
The authors have no relevant affiliations or financial involvement with any organization or entity with a financial interest in or financial conflict with the subject matter or materials discussed in the manuscript. This includes employment, consultancies, honoraria, stock ownership or options, expert testimony, grants or patents received or pending, or royalties.

DATA AVAILABILITY STATEMENT
All data used to create the figures in this article are available from the scpdata package: https://bioconductor.org/packages/release/data/experiment/html/scpdata.html. The R code to reproduce the figures is available at: https://github.com/UCLouvain-CBIO/2022-scp-data-analysis.

LITERATURE CITED
Amezquita, R. A., Lam, A. T. L., Becht, E., Carey, V. J., Carpp, L. N., Geistlinger, L., … Hicks, S. C. (2020). Orchestrating single-cell analysis with bioconductor. Nature Methods, 17(2), 137–145. doi: 10.1038/s41592-019-0654-x
Andrecut, M. (2009). Parallel GPU implementation of iterative PCA algorithms. Journal of Computational Biology, 16(11), 1593–1599. doi: 10.1089/cmb.2008.0221
Argelaguet, R., Cuomo, A. S. E., Stegle, O., & Marioni, J. C. (2021). Computational principles and challenges in single-cell data integration. Nature Biotechnology, 39(10), 1202–1215. doi: 10.1038/s41587-021-00895-7
Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics, 2008(10), P10008. doi: 10.1088/1742-5468/2008/10/P10008
Boekweg, H., Watt, D. V. D., Tuong, T., Madisyn Johnston, S., Guise, A. J., Plowey, E. D., … Payne, S. H. (2022). Features of peptide fragmentation spectra in Single-Cell proteomics. Journal of Proteome Research, 21(1), 182–188. doi: 10.1021/acs.jproteome.1c00670
Bramer, L. M., Irvahn, J., Piechowski, P. D., Rodland, K. D., & Webb-Robertson, B. J. M. (2021). A review of imputation strategies for isobaric labeling-based shotgun proteomics. Journal of Proteome Research, 20(1), 1–13. doi: 10.1021/acs.jproteome.0c00123
Brunner, A.-D., Thielert, M., Vasilopoulou, C., Ammar, C., Coscia, F., Mund, A., … Mann, M.
Cuklina, J., Lee, C. H., Williams, E. G., Sajic, T., Ctortecka, C., Stejskal, K., Krssakova, G., Mend-... & Mechtler, K. (2021). The rise of single-cell proteomics workflow identifies >1000 protein groups per mammalian cell. Chemical Science, 12(3), 1001–1006. doi: 10.1039/d0sc03636f

Cox, J., Hein, M. Y., Luber, C. A., Paron, I., Nagaraj, N., & Mann, M. (2014). Accurate proteome-wide label-free quantification by delay normalization and maxima peptide ratio extraction, termed MaxLFQ. Molecular & Cellular Proteomics, 13(9), 2513–2526. doi: 10.1074/mcp.M113.031591

Crowell, H. L., Soneson, C., Germain, P.-L., Calini, D., Collin, L., Raposo, C., … Robinson, M. D. (2020). Muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data. Nature Communications, 11(1), 6077. doi: 10.1038/s41467-020-19894-4

Ctortecka, C., & Mechtler, K. (2021). The rise of single-cell proteomics. Analytical Science, 2(3-4), 84–94. doi: 10.1002/ansa.202000152

Ctortecka, C., Krssakova, G., Stejskal, K., Penninger, J. M., Mendjan, S., Mechtler, K., & Stadlmann, J. (2021). Comparative proteome signatures of trace samples by multiplexed data-independent acquisition. Molecular & Cellular Proteomics, 21(1), 100177. doi: 10.1016/j.mcpro.2021.100177

Ctortecka, C., Stejskal, K., Krssakova, G., Mendjan, S., & Mechtler, K. (2022). Quantitative accuracy and precision in multiplexed single-cell proteomics. Analytical Chemistry, 94(5), 2434–2443. doi: 10.1021/acs.analchem.1c04174

Cuklina, J., Lee, C. H., Williams, E. G., Sajic, T., Collins, B. C., Martinez, M. R., … Pedrioli, P. G. A. (2021). Diagnostics and correction of batch effects in large-scale proteomic studies: A tutorial. Molecular Systems Biology, 17(8), e10240. doi: 10.15252/msb.202110798

Budnik, B., Levy, E., Harmange, G., & Slavov, N. (2018). SCoPE-MS: Mass spectrometry of single mammalian cells quantifies proteome heterogeneity during cell differentiation. Genome Biology, 19(1), 161. doi: 10.1186/s13059-018-1547-5

Chen, A. T., Franks, A., & Slavov, N. (2019). DART-ID increases single-cell proteome coverage. PLOS Computational Biology, 15(7), e1007082. doi: 10.1371/journal.pcbi.1007082

Cheung, T. K., Lee, C.-Y., Bayer, F. P., McCoy, A., Kuster, B., & Rose, C. M. (2021). Defining the carrier proteome limit for single-cell proteomics. Nature Methods, 18(1), 76–83. doi: 10.1038/s41592-020-01002-5

Cong, Y., Liang, Y., Motamedchaboki, K., Huguet, R., Truong, T., Zhao, R., … Kelly, R. T. (2020). Improved Single-Cell Proteome coverage using narrow-bore packed NanoLC columns and ultrasensitive mass spectrometry. Analytical Chemistry, 92(3), 2665–2671. doi: 10.1021/acs.analchem.9b04631

Cong, Y., Motamedchaboki, K., Misal, S. A., Liang, Y., Guise, A. J., Truong, T., … Kelly, R. T. (2021). Ultrasonic single-cell proteomics: a high-throughput single-cell proteomics tool. Analytical Chemistry, 93(10), 6278–6287. doi: 10.1021/acs.analchem.9b00780

Derks, J., Leduc, A., Wallmann, G., Gray Huffman, R., Willetts, M., Khan, S., … Slavov, N. (2022). Increasing the throughput of sensitive proteomics by plexDIA. Nature Biotechnology, doi: 10.1038/s41587-022-01389-w

Dou, M., Clair, G., Tsai, C.-F., Xu, K., Chrisler, W. B., Sontag, R. L., … Zhu, Y. (2019). High-Throughput single cell proteomics enabled by multiplex iSCARIC labeling in a nanodroplet sample preparation platform. Analytical Chemistry, 91(20), 13119–13127. doi: 10.1021/acs.analchem.9b03349

Fondrie, W. E., & Noble, W. S. (2020). Machine learning strategy that leverages large data sets to boost statistical power in small-scale experiments. Journal of Proteome Research, 19(3), 1267–1274. doi: 10.1021/acs.jproteome.9b00780

Fondrie, W. E., & Noble, W. S. (2021). Mokapot: Fast and flexible semisupervised learning for peptide detection. Journal of Proteome Research, 20(4), 1966–1971. doi: 10.1021/acs.jproteome.0c01010

Gardner, P. P., Paterson, J. M., McGimpsey, S., Ashari-Ghomi, F., Umu, S. U., Pawlik, A., … Black, M. A. (2022). Sustained software development, not number of citations or journal choice, is indicative of accurate bioinformatic software. Genome Biology, 23(1), 56. doi: 10.1186/s13059-022-02625-x

Gatto, L., Aebersold, R., Cox, J., Demichev, V., Derks, J., Emmott, E., … Slavov, N. (2022). Initial recommendations for performing, benchmarking, and reporting single-cell proteomics experiments. ArXiv, Available at https://arxiv.org/ftp/arxiv/papers/2207/2207.10815.pdf

Gebreyesus, S. T., Siyal, A. A., Kitata, R. B., Chen, E. S.-W., Angkhambar, B., Angata, T., … Tu, H.-L. (2023). Streamlined single-cell proteomics by an integrated microfluidic chip and data-independent acquisition mass spectrometry. Nature Communications, 14(1), 37. doi: 10.1038/s41467-021-27778-4

Germain, P. L., Sonrel, A., & Robinson, M. D. (2020). Pipecomp, a general framework for the evaluation of computational pipelines, reveals performant single cell RNA-seq preprocessing tools. Genome Biology, 21(1), 227. doi: 10.1186/s13059-020-02136-7

Goeminne, L. J. E., Sticker, A., Martens, L., Gevaert, K., & Clement, L. (2020). MSqRob takes the missing hurdle: Uniting intensity- and count-based proteomics. Analytical Chemistry, 92(9), 6278–6287. doi: 10.1021/acs.analchem.9b04375

Gray Huffman, R., Leduc, A., Wichmann, C., Gioia, M., Borriello, F., Specht, H., … Slavov, N. (2022). Prioritized single-cell proteomics reveals molecular and functional polarization across primary macrophages. BioRxiv, Available at: https://www.biorxiv.org/content/10.1101/2022.03.16.484655v1.full
Vanderaa and Gatto
18 of 19

Hartlimayr, D., Ctortecka, C., Seth, A., Mendjan, S., Tourniaire, G., & Mechtler, K. (2021). An automated workflow for label-free and multiplexed single cell proteomics sample preparation at unprecedented sensitivity. BioRxiv, Available at https://www.biorxiv.org/content/10.1101/2021.04.14.439828v1

Huber, W., Carey, V. J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B. S., … Morgan, M. (2015). Orchestrating high-throughput genomic analysis with bioconductor. Nature Methods, 12(2), 115–121. doi: 10.1038/nmeth.3252

Johnson, W. E., Li, C., & Rabinovic, A. (2007). Adjusting batch effects in microarray expression data using empirical bayes methods. Biostatistics, 8(1), 118–127. doi: 10.1093/biostatistics/kxj037

Kalxdorf, M., Miller, T., Stegle, O., & Krijgsveld, J. (2021). IceR improves proteome coverage and data completeness in global and single-cell proteomics. Nature Communications, 12(1), 4787. doi: 10.1038/s41467-021-25077-6

Kelly, R. T. (2020). Single-cell proteomics: Progress and prospects. Molecular & Cellular Proteomics, 19(11), 1739–1748. doi: 10.1074/mcp.R120.002234

Lazar, C., Gatto, L., Ferro, M., Bruley, C., & Burger, T. (2016). Accounting for the multiple natures of missing values in label-free quantitative proteomics data sets to compare imputation strategies. Journal of Proteome Research, 15(4), 1116–1125. doi: 10.1021/acs.jproteome.5b00981

Leduc, A., Huffman, R. G., & Slavov, N. (2022). Droplet sample preparation for single-cell proteomics applied to the cell cycle. BioRxiv, Available at: https://www.biorxiv.org/content/10.1101/2021.04.24.441211v1

Levy, E., & Slavov, N. (2018). Single cell protein analysis for systems biology. Essays in Biochemistry, 62(4), 595–605. doi: 10.1042/EBC20180014

Li, Y., Li, H., Xie, Y., Chen, S., Qin, R., Dong, H., … Qin, W. (2021). An integrated strategy for mass spectrometry-based multimomics analysis of single cells. Analytical Chemistry, 93(42), 14059–14067. doi: 10.1021/acs.analchem.0c05209

Liang, Y., Acor, H., McCown, M. A., Nwosu, A. J., Boekweg, H., Axtell, N. B., … Kelly, R. T. (2021). Fully automated sample processing and analysis workflow for Low-Input proteome profiling. Analytical Chemistry, 93(3), 1658–1666. doi: 10.1021/acs.analchem.0c04240

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., & Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. Nature Methods, 15(12), 1053–1058. doi: 10.1038/s41592-018-0229-2

Lun, A. T. L., & Marioni, J. C. (2017). Overcoming confounding plate effects in differential expression analyses of single-cell RNA-seq data. Biostatistics, 18(3), 451–464. doi: 10.1093/biostatistics/kxx055

Mereu, E., Laﬁzi, A., Moutinho, C., Ziegengainh, C., McCarthy, D. J., Alvarez-Varela, A., … Heyn, H. (2020). Benchmarking single-cell RNA-sequencing protocols for cell atlas projects. Nature Biotechnology, 38(6), 747–755. doi: 10.1038/s41587-020-0469-4

Orsburn, B. C., Yuan, Y., & Bumpus, N. N. (2022). Single cell proteomics using a trapped ion mobility Time-of-Flight mass spectrometer provides insight into the post-translational modiﬁcation landscape of individual human cells. BioRxiv, Available at: https://www.biorxiv.org/content/10.1101/2022.02.12.480144v1

Petelski, A. A., Emmott, E., Leduc, A., Gray Huffman, R., Specht, H., Perlman, D. H., & Slavov, N. (2021). Multiplexed single-cell proteomics using SCoPE2. Nature Protocols, 16(12), 5398–5425. doi: 10.1038/s41596-021-00616-2

Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., & Vert, J. P. (2018). A general and flexible method for signal extraction from single-cell RNA-seq data. Nature Communication, 9(1), 284. doi: 10.1038/s41467-017-02554-5

Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., & Smyth, G. K. (2015). limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Research, 43(7), e47. doi: 10.1093/nar/gkv007

Schoof, E. M., Furtwngler, B., Resin, N., Rapin, N., Savickas, S., Gentil, C., … Porse, B. T. (2021). Quantitative single-cell proteomics as a tool to characterize cellular hierarchies. Nature Communications, 12(1), 745679. doi: 10.1038/s41467-021-23667-y

Slavov, N. (2020). Unpicking the proteome in single cells. Science, 367(6477), 512–513. doi: 10.1126/science.aaz6695

Slavov, N. (2021a). Driving single cell proteomics forward with innovation. Journal of Proteome Research, 20(11), 4915–4918. doi: 10.1021/acs.jproteome.1c00639

Slavov, N. (2021b). Increasing proteomics throughput. Nature Biotechnology, 39(7), 809–810. doi: 10.1038/s41587-021-00881-z

Slavov, N. (2021c). Scaling up single-cell proteomics. Molecular & Cellular Proteomics, 21(1), 100179. doi: 10.1016/j.mcp.2021.100179

Slavov, N. (2021d). Single-cell protein analysis by mass spectrometry. Current Opinion in Chemical Biology, 60, 1–9. doi: 10.1016/j.cbpa.2020.04.018

Slavov, N. (2022). Learning from natural variation across the proteomes of single cells. PLOS Biology, 20(1), e3001512. doi: 10.1371/journal.pbio.3001512

Specht, H., & Slavov, N. (2021). Optimizing accuracy and depth of protein quantification in experiments using isobaric carriers. Journal of
Expert Review
Nature Communications
Cell Systems
BioRxiv
Journal
Nature Communications, Analytical Chemistry
Current Protocols
Analytical Chemistry
Molecular & Cellular Proteomics
10.1101/2021.08.17.456676v2
19 of 19

Webber, K. G. I., Truong, T., Madisyn Johnston, S., Zapata, S. E., Liang, Y., Davis, J. M., ... Kelly, R. T. (2022). Label-free profiling of up to 200 single-cell proteomes per day using a dual-column nanoflow liquid chromatography platform. Analytical Chemistry, 94(15), 6017–6025. doi: 10.1021/acs.analchem.2c00646

Williams, S. M., Liyu, A. V., Tsai, C.-F., Moore, R. J., Orton, D. J., Chrisler, W. B., ... Zhu, Y. (2020). Automated coupling of nanodroplet sample preparation with liquid chromatography-mass spectrometry for high-throughput single-cell proteomics. Analytical Chemistry, 92(15), 10588–10596. doi: 10.1021/acs.analchem.0c01551

Wolf, F. A., Angerer, P., & Theis, F. J. (2018). SCANPY: Large-scale single-cell gene expression data analysis. Genome Biology, 19(1), 15. doi: 10.1186/s13059-017-1382-0

Woo, J., Williams, S. M., Markillie, L. M., Feng, S., Tsai, C.-F., Aguiler-Vazquez, V., ... Zhu, Y. (2021). High-throughput and high-efficiency sample preparation for single-cell proteomics using a nested nanowell chip. Nature Communications, 12(1), 6246. doi: 10.1038/s41467-021-26514-2

Woo, J., Clair, G. C., Williams, S. M., Moore, R. J., ... Zhu, Y. (2022). Three-dimensional feature matching improves coverage for single-cell proteomics based on ion mobility filtering. Cell Systems, 13(5), 426–434.e4.

Yu, S.-H., Kyriakidou, P., & Cox, J. (2020). Isobaric matching between runs and novel PSM-level normalization in MaxQuant strongly improve reporter ion-based quantification. Journal of Proteome Research, 19(10), 3945–3954. doi: 10.1021/acs.jproteome.0c00209

Zhu, Y., Clair, G., Chrisler, W. B., Shen, Y., Zhao, R., Shukla, A. K., ... Kelly, R. T. (2018a). Proteomic analysis of single mammalian cells enabled by microfluidic nanodroplet sample preparation and ultrasensitive NanoLC-MS. Angewandte Chemie International Edition in English, 57(38), 12370–12374. doi: 10.1002/anie.201802843

Zhu, Y., Piehowski, P. D., Zhao, R., Chen, J., Shen, Y., Moore, R. J., ... Kelly, R. T. (2018b). Nanodroplet processing platform for deep and quantitative proteome profiling of 10-100 mammalian cells. Nature Communications, 9(1), 882. doi: 10.1038/s41467-018-03367-w

Zhu, Y., Scheibinger, M., Ellwanger, D. C., Krey, J. F., Choi, D., Kelly, R. T., ... Barr-Gillespie, P. G. (2019). Single-cell proteomics reveals changes in expression during hair-cell development. Elife, 8, e50777. doi: 10.7554/eLife.50777