Standardised and reproducible analysis of mass spectrometry-based single-cell proteomics data

Replication of the SCoPE2 analysis by Specht et al. 2019

Christophe Vanderaa, Laurent Gatto

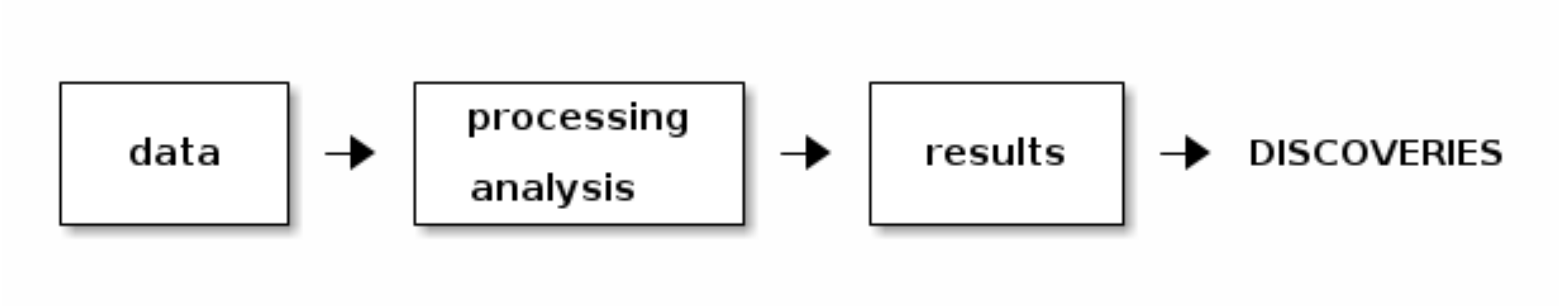
Computational Biology Unit (CBIO), de Duve Institute, UCLouvain

18 August 2020

Outline

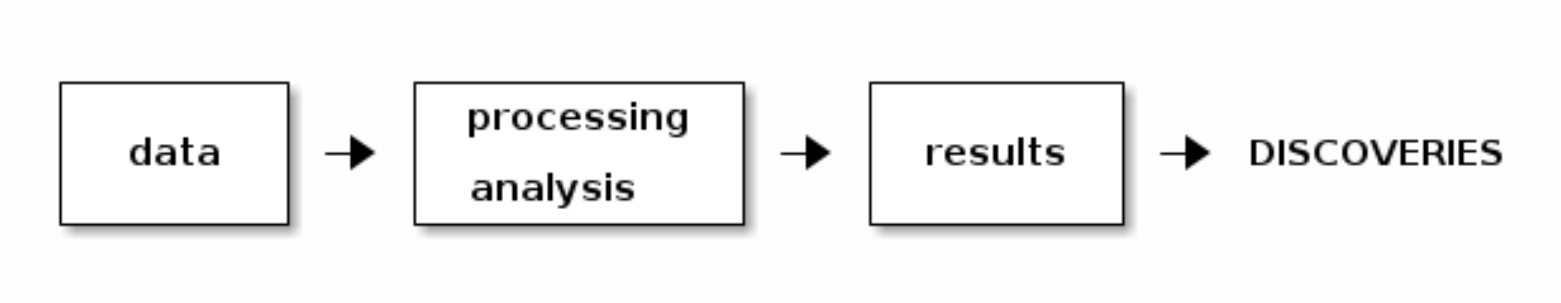
The value of replication (1)

- Expectation

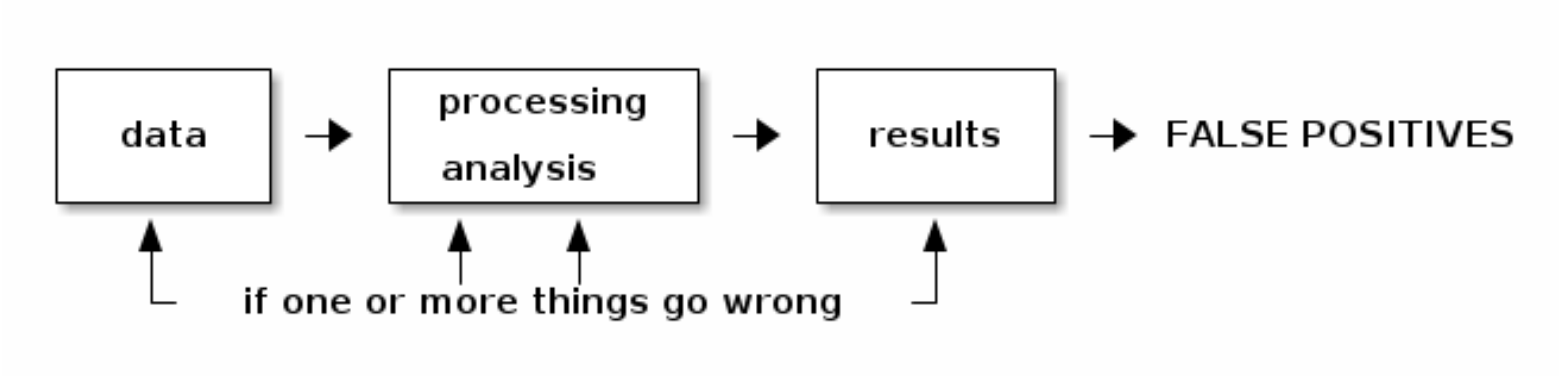


The value of replication (1)

- Expectation



- Reality



The value of replication (2)

- **Reproduction-based development** agreement between the developer, the data producer and the user.

Replication is the first step to define sound data infrastructure and principled analysis.

Why SCoPE2

- SCoPE2 quantifies thousands of proteins x thousands single-cells
- Full protocole available
- Full analysis script and data available

Our objectives

- Contribute a standardized and principled data and analysis that is broadly applicable.
- Reproducible computational infrastructure to further improve data analysis and interpretation.
- R/Bioconductor is an ideal environment to attain these goals.

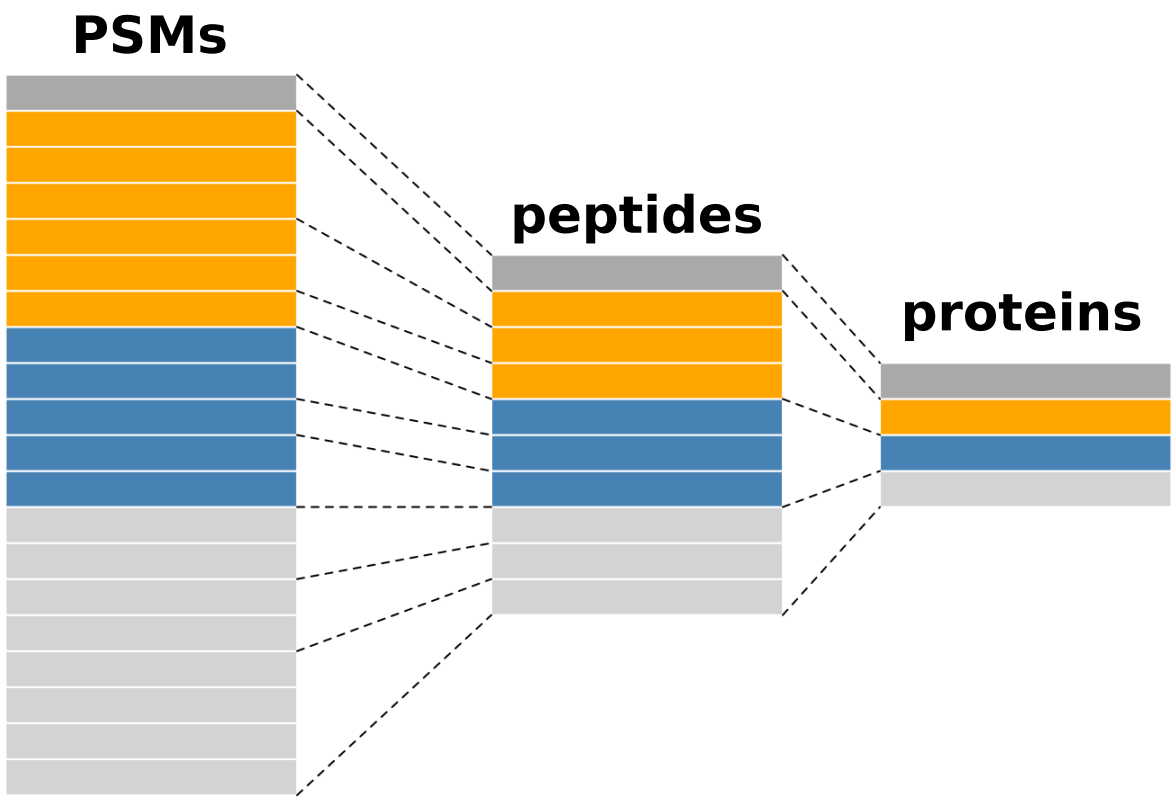
Implemented in the `scp` package.

Outline

Data infrastructure: QFeatures[1]

scp package

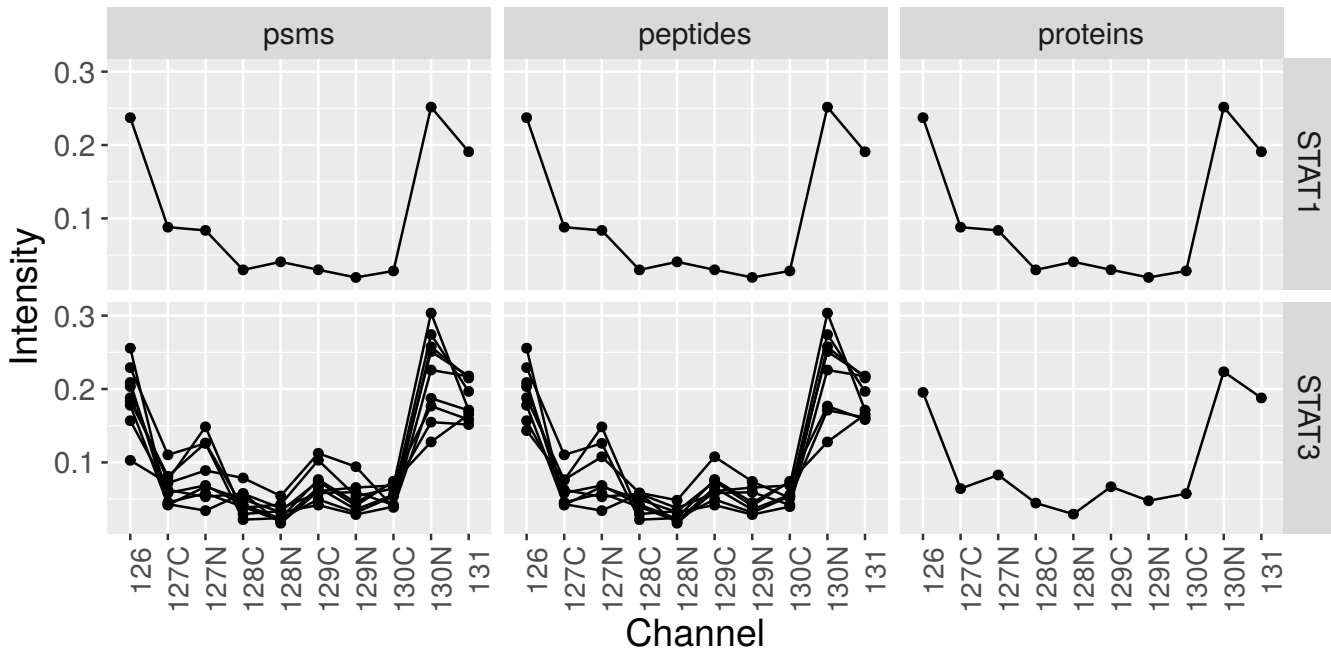
`QFeatures`: data framework dedicated to manipulate and process MS-based quantitative data.



[1] Gatto (2020)

Data infrastructure: QFeatures[1]

`QFeatures`: data framework dedicated to manipulate and process MS-based quantitative data.

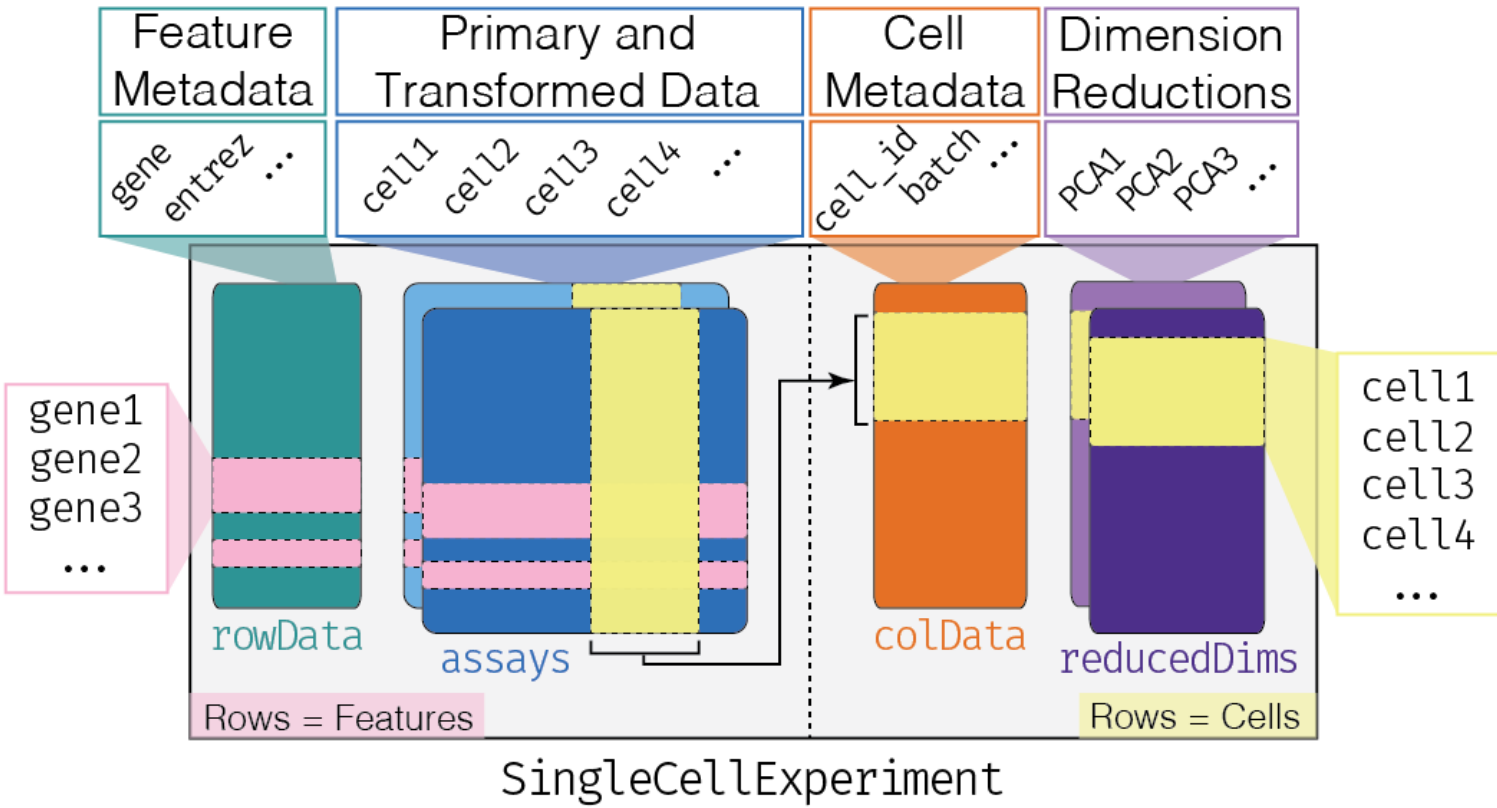


[1] Gatto (2020)

Data infrastructure: SingleCellExperiment[2,3]

scp package

`SingleCellExperiment`: provides dedicated framework for single-cell data analysis.

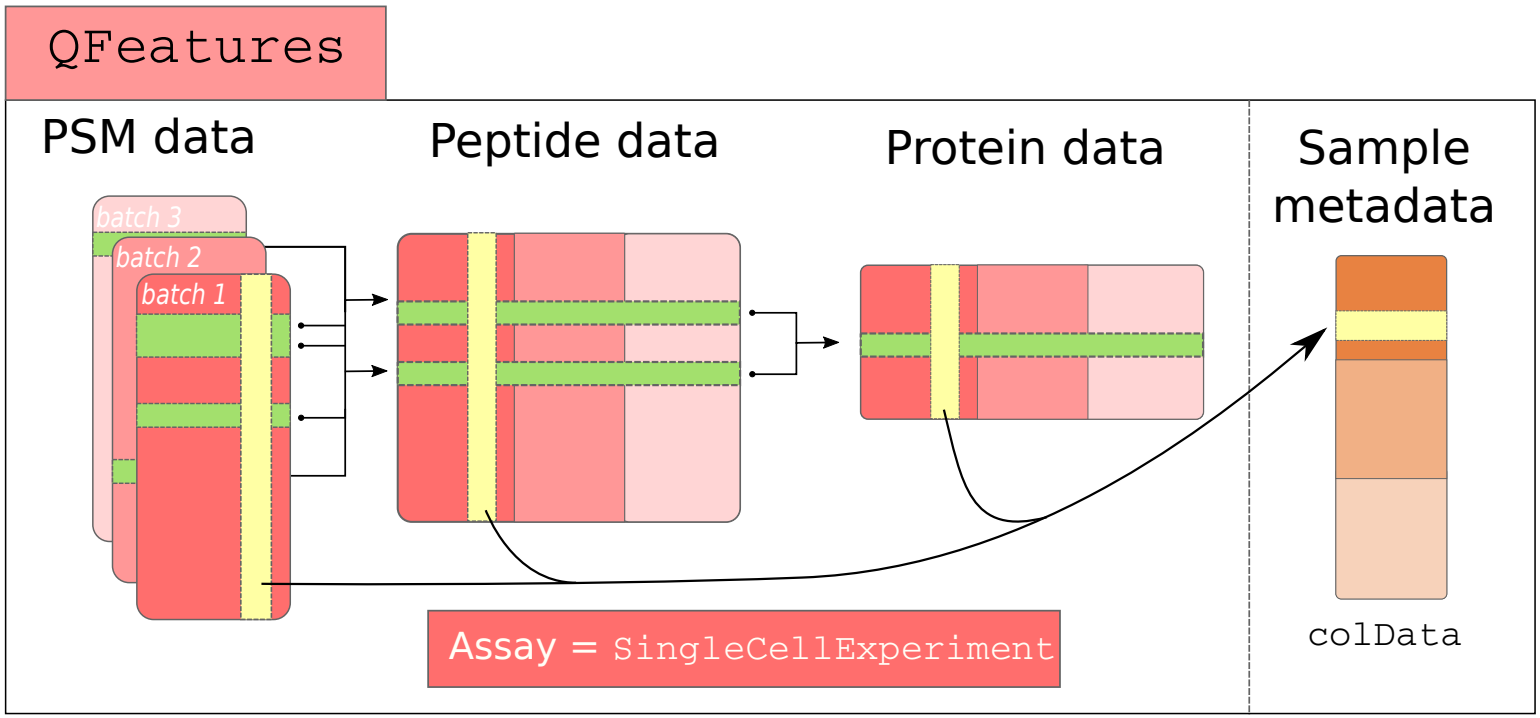


[2] Lun and Risso (2020)

[3] Amezquita et al. (2019)

Data infrastructure: scp

`scp` = `SingleCellExperiment` + `QFeatures`



Load data

Load the SCoPE2 dataset called `specht2019v2`[4]

```
library(scpdata)
data("specht2019v2")
```

Dataset overview

```
show(specht2019v2)
```

```
An instance of class QFeatures containing 179 assays:
 [1] 190222S_LCA9_X_FP94AA: SingleCellExperiment with 2823 rows and 11 col...
 [2] 190222S_LCA9_X_FP94AB: SingleCellExperiment with 4297 rows and 11 col...
 [3] 190222S_LCA9_X_FP94AC: SingleCellExperiment with 4956 rows and 11 col...
 ...
 [177] 191110S_LCB7_X_APNOV16plex2_Set_9: SingleCellExperiment with 4626 r...
 [178] peptides: SingleCellExperiment with 9208 rows and 1018 columns
 [179] proteins: SingleCellExperiment with 2772 rows and 1018 columns
```

Tabular data (such as generated from MaxQuant, ProteomeDiscoverer, ...) can be read using the `readSCP()` function.

[4] Specht et al. (2019)

Metadata

`colData` stores sample metadata for **all assays** in one table

Set	Channel	SampleType	lcbatch	sortday	digest
190222S_LCA9_X_FP94AA	RI1	Carrier	LCA9	s8	N
190222S_LCA9_X_FP94AA	RI2	Reference	LCA9	s8	N
190222S_LCA9_X_FP94AA	RI3	Unused	LCA9	s8	N
190222S_LCA9_X_FP94AA	RI4	Macrophage	LCA9	s8	N
190222S_LCA9_X_FP94AA	RI5	Macrophage	LCA9	s8	N
190222S_LCA9_X_FP94AA	RI6	Macrophage	LCA9	s8	N
...	...	...	...	...	...

Analysis workflow

1. Load data

PSM data

```
[1] 190222S_LCA9_X_FP94AA: SingleCellExperiment with 2823 rows and 11 columns
[2] 190222S_LCA9_X_FP94AB: SingleCellExperiment with 4297 rows and 11 columns
[3] 190222S_LCA9_X_FP94AC: SingleCellExperiment with 4956 rows and 11 columns
...
[177] 191110S_LCB7_X_APNOV16plex2_Set_9: SingleCellExperiment with 4626 rows and 16 columns
```

Analysis workflow

1. Load data

PSM data

```
[1] 190222S_LCA9_X_FP94AA: SingleCellExperiment with 2823 rows and 11 columns
[2] 190222S_LCA9_X_FP94AB: SingleCellExperiment with 4297 rows and 11 columns
[3] 190222S_LCA9_X_FP94AC: SingleCellExperiment with 4956 rows and 11 columns
...
[177] 191110S_LCB7_X_APNOV16plex2_Set_9: SingleCellExperiment with 4626 rows and 16 columns
```

2. PSM filtering
3. Expression channel by reference channel division
4. PSM to peptides aggregating
5. Single cells filtering based on median CV
6. Normalization
7. Removal of highly missing peptides
8. Log-transformation

Peptide data

```
[178] peptides: SingleCellExperiment with 9208 rows and 1018 columns
```

Analysis workflow

1. Load data

PSM data

```
[1] 190222S_LCA9_X_FP94AA: SingleCellExperiment with 2823 rows and 11 columns
[2] 190222S_LCA9_X_FP94AB: SingleCellExperiment with 4297 rows and 11 columns
[3] 190222S_LCA9_X_FP94AC: SingleCellExperiment with 4956 rows and 11 columns
...
[177] 191110S_LCB7_X_APNOV16plex2_Set_9: SingleCellExperiment with 4626 rows and 16 columns
```

2. PSM filtering
3. Expression channel by reference channel division
4. PSM to peptides aggregating
5. Single cells filtering based on median CV
6. Normalization
7. Removal of highly missing peptides
8. Log-transformation

Peptide data

```
[178] peptides: SingleCellExperiment with 9208 rows and 1018 columns
```

9. Peptides to proteins aggregation
10. Normalization
11. Imputation
12. Batch correction

Peptide data

```
[179] proteins: SingleCellExperiment with 2772 rows and 1018 columns
```

Analysis workflow

1. Load data

PSM data

```
[1] 190222S_LCA9_X_FP94AA: SingleCellExperiment with 2823 rows and 11 columns
[2] 190222S_LCA9_X_FP94AB: SingleCellExperiment with 4297 rows and 11 columns
[3] 190222S_LCA9_X_FP94AC: SingleCellExperiment with 4956 rows and 11 columns
...
[177] 191110S_LCB7_X_APNOV16plex2_Set_9: SingleCellExperiment with 4626 rows and 16 columns
```

2. **PSM filtering**
3. Expression channel by reference channel division
4. PSM to peptides aggregating
5. **Single cells filtering based on median CV**
6. Normalization
7. **Removal of highly missing peptides**
8. **Log-transformation**

Peptide data

```
[178] peptides: SingleCellExperiment with 9208 rows and 1018 columns
```

9. **Peptides to proteins aggregation**
10. Normalization
11. Imputation
12. **Batch correction**

Peptide data

```
[179] proteins: SingleCellExperiment with 2772 rows and 1018 columns
```

Outline

PSM filtering

Filter out features based on the feature metadata

Example: filter out reverse hits. The filter is applied to the `Reverse` field in the feature metadata

```
filterFeatures(specht2019v2,
               ~ Reverse != "+")
```

Source code in `QFeatures`

QC metrics (1)

Interesting metrics for MS-SCP quality control:

- Sample to carrier ratio: ratio of the carrier channel intensity signal over the sample channel intensity
- Peptide FDR[5]: expected rate of wrongly assigned features to a given peptide
- Cell median CV[6]: reliability of the protein quantification summarized over each cell.

Example:

```
computeMedianCV(specht2019v2,
                i = "peptides",
                proteinCol = "protein",
                peptideCol = "peptide",
                batchCol = "Set")
```

Source code in `scp`

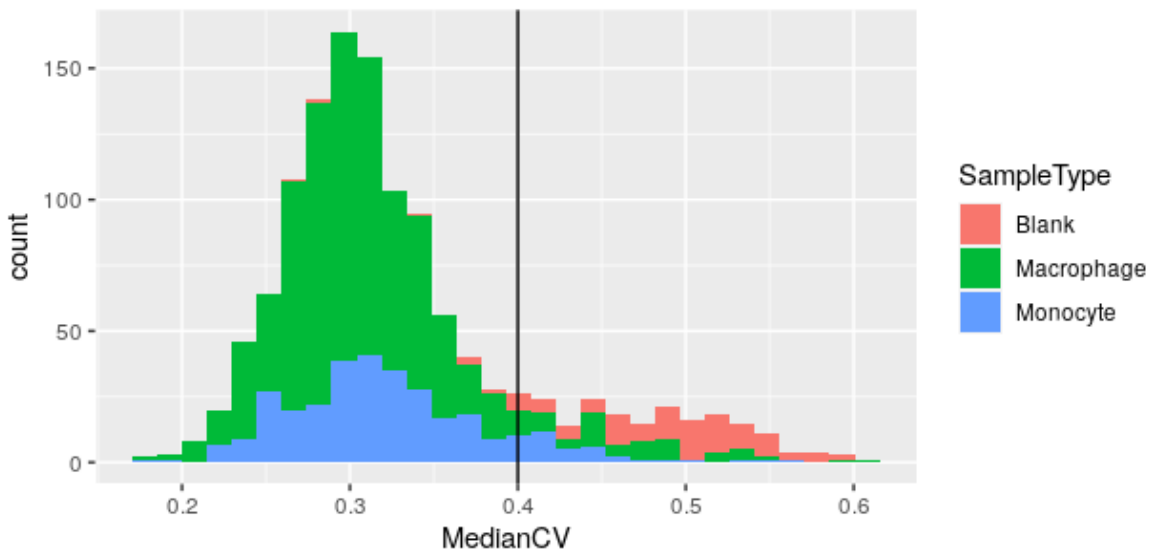
[5] false discovery rate

[6] coefficient of variation

QC metrics (2)

QC metrics are stored in the data set for plotting or subsetting

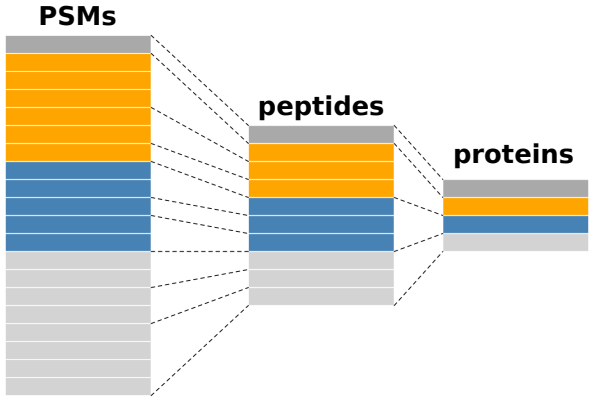
```
library(tidyverse)
specht2019v2[["peptides"]] %>%
  colData %>%
  data.frame %>%
  ggplot(aes(x = MedianCV,
             fill = SampleType)) +
  geom_histogram() +
  geom_vline(xintercept = 0.4)
```



Feature aggregation

Feature aggregation includes 2 steps:

- Combine the quantiative data from multiple features to a single aggregated features
- Store the relationship between the parent features and the aggregated features



Example: aggregate peptides to proteins

```
aggregateFeatures(specht2019v2,
                  i = "peptides",
                  name = "proteins",
                  fcol = "protein",
                  fun = colMedians, na.rm = TRUE)
```

Source code in `QFeatures`

Managing missingness

`0`'s can be either **biological** or **technical** zero. They are better relaced by `NA`'s.

```
zeroIsNA(specht2019v2,
         i = "peptides")
```

Features containing too many missing data (e.g. $>= 99$ %) should be removed

```
filterNA(specht2019v2,
         i = "peptides",
         pNA = 0.99)
```

Source code in `QFeatures`

Data transformation

Common data transformation can easily be applied:

- Normalization
- Log-transformation
- Imputation

Example: log_2-transformation:

```
logTransform(specht2019v2,
             i = "peptides",
             base = 2,
             name = "peptides_log")
```

Source code in `QFeatures`

Custom functions

Some custom function can be applied to the data set too. Example: batch correction using `sva::ComBat`. First, extract the data

```
sce <- specht2019v2[["proteins"]]
```

Build the correction matrix and apply the ComBat algorithm

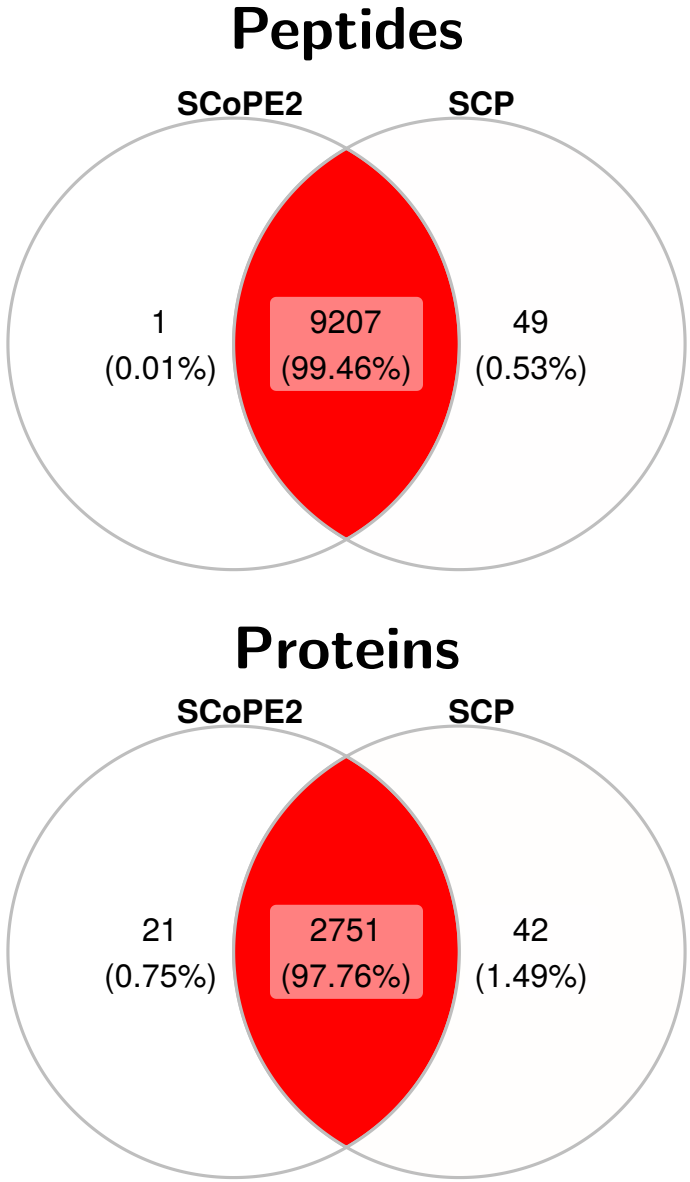
```
batch <- colData(sce)$Set
model <- model.matrix(~ SampleType, data = colData(sce))
assay(sce) <- ComBat(dat = assay(sce),
                     batch = batch,
                     mod = model)
```

Add the corrected protein to the dataset and keep feature relationships

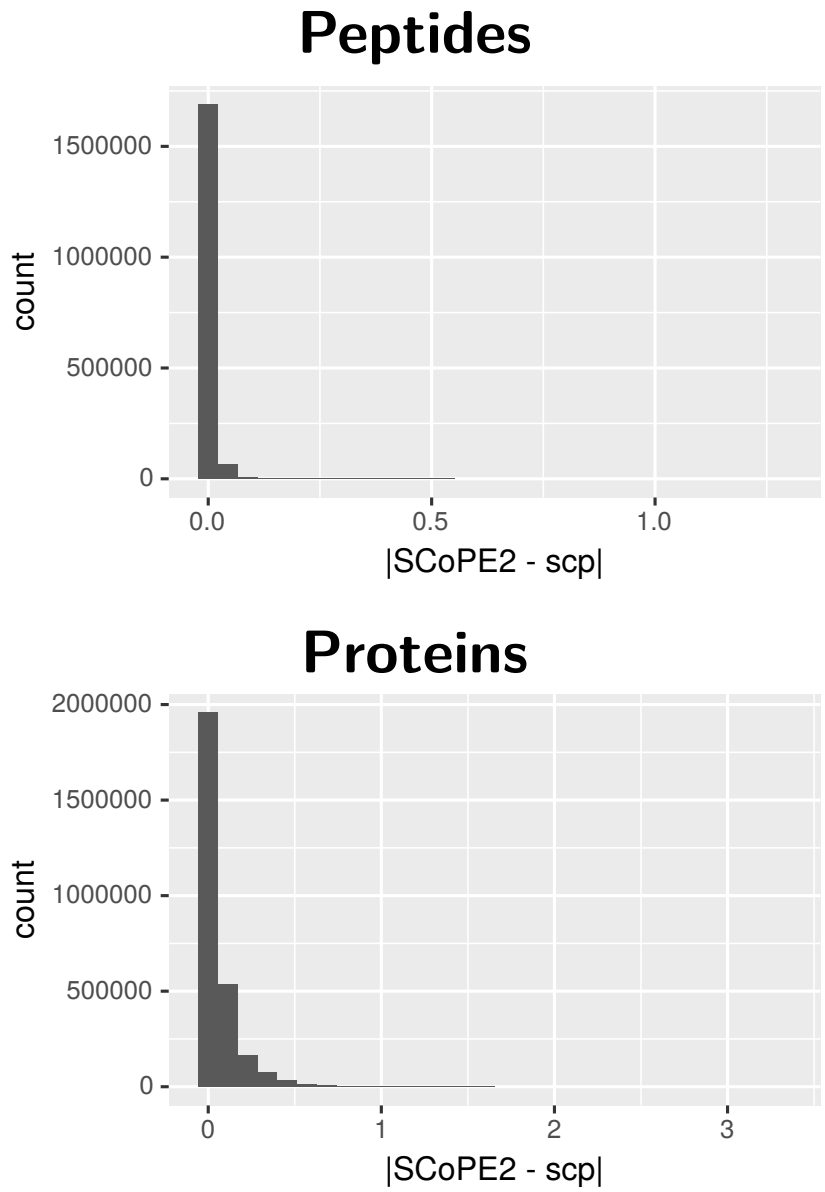
```
addAssay(specht2019v2,
         sce,
         name = "proteins_batchC") %>%
addAssayLinkOneToOne(from = "proteins",
                     to = "proteins_batchC")
```

Outline

Selected features

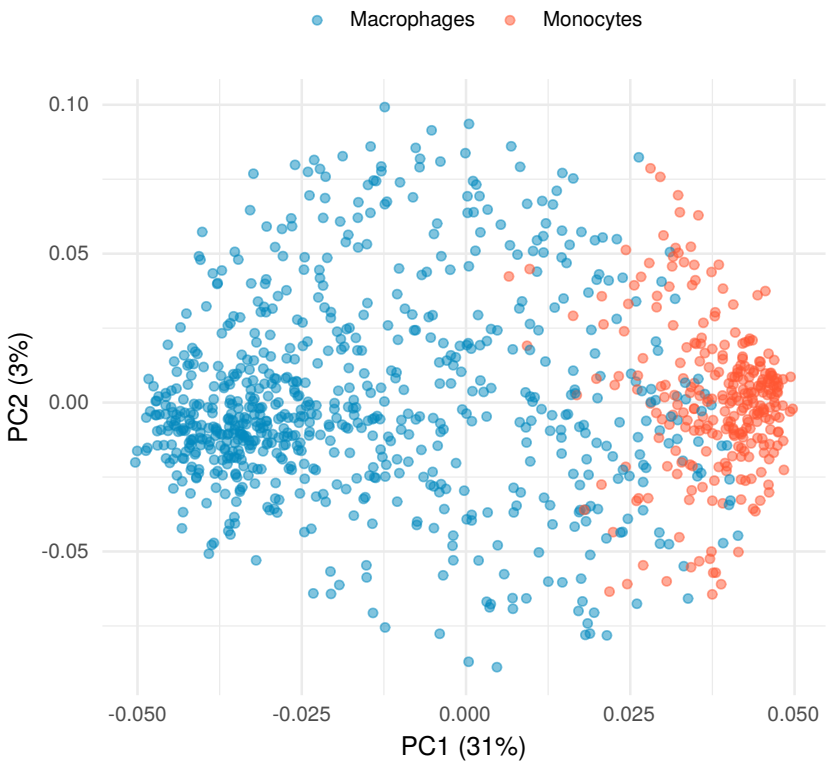


Numerical comparison

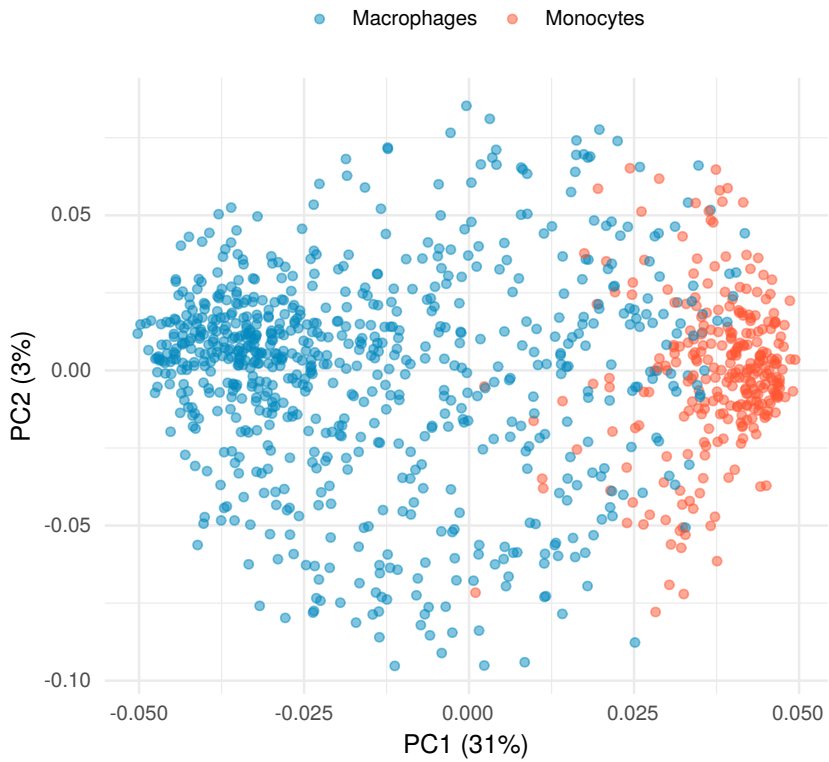


Replicate weighted PCA

SCoPE2



scp

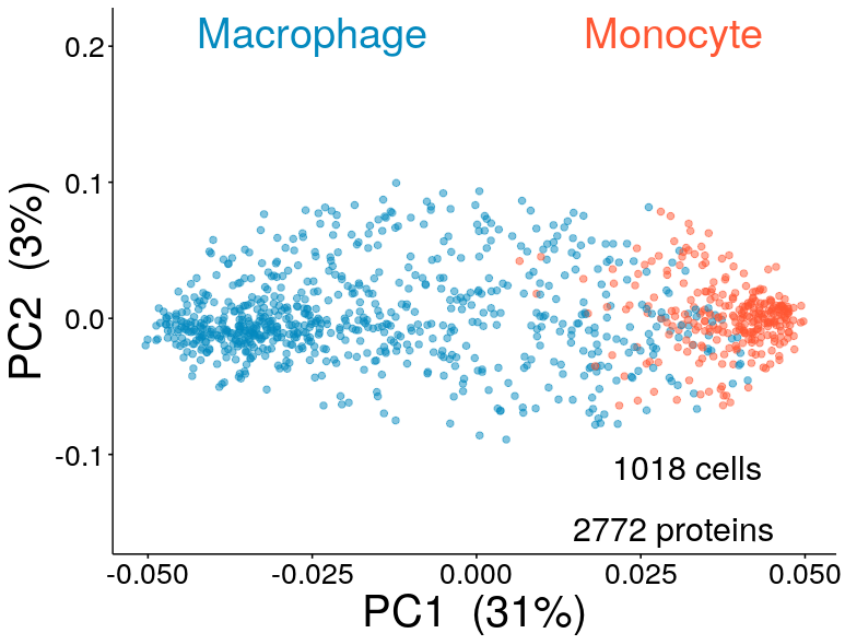


Note about replication

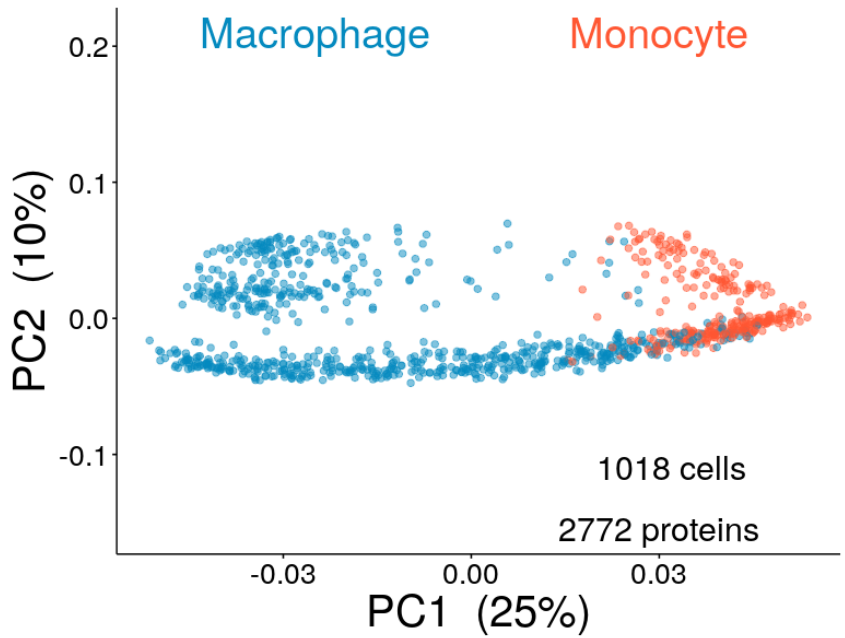
Versioning is essential for replication

Example: batch correction using the `ComBat` algorithm from `sva`

`sva 3.34.0`



`sva 3.36.0`

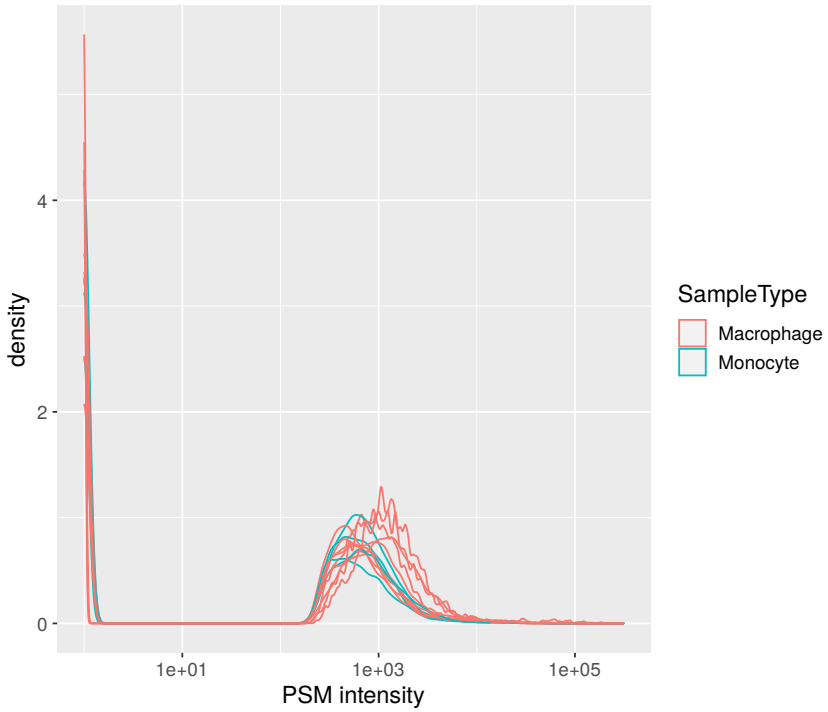


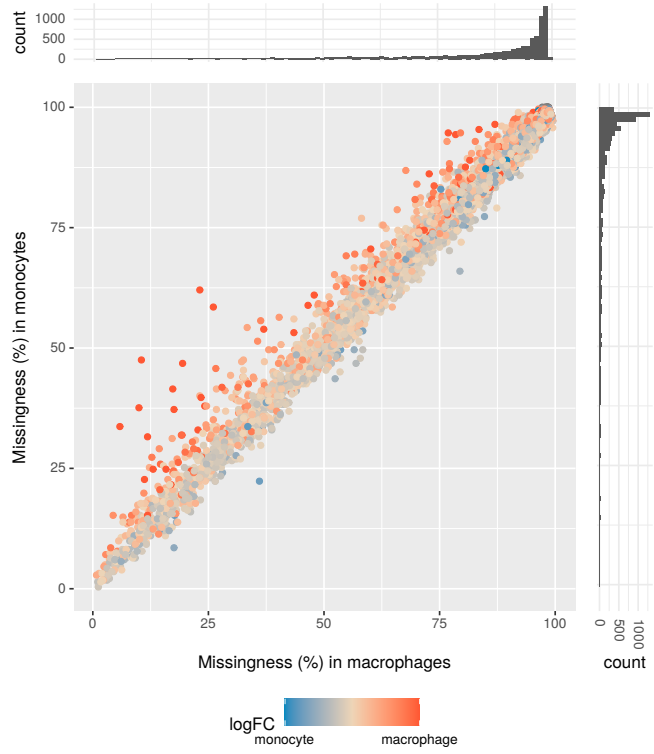
Outline

Take home message

- MS-based single cell proteomics: young field, with many challenges and great progess. `scp` to address the need for principled and reproducible data analysis.
- `scp` isn't specific to SCoPE2/TMT data, applicable to other LF protocols such as nanoPOTS (Williams et al. (2020); Cong et al. (2020)).
- `scp` and `SingleCellExperiment`: same infrastructure for single cell proteomics and RNA sequencing.
- Tool for novel computational developments.

Future directions





Resources

- `scp`: `http://UClouvain-CBIO.github.io/scp`
- `scpdata`: coming soon
- `QFeatures`: `http://rformassspectrometry.org`
- `SingleCellExperiment`: Bioconductor
- **Slides**: `http://bit.ly/2020SCP` under CC-BY SA

Resources

- `scp`: `http://UClouvain-CBIO.github.io/scp`
- `scpdata`: coming soon
- `QFeatures`: `http://rformassspectrometry.org`
- `SingleCellExperiment`: Bioconductor
- **Slides**: `http://bit.ly/2020SCP` under CC-BY SA

Acknowledgements

- Nikolai Slavov, Harrison Specht, Ed Emmott.
- Fonds National de la Recherche Scientifique (FNRS)
- **Thank you for your attention**

References I

Robert A Amezquita, Aaron T L Lun, Etienne Becht, Vince J Carey, Lindsay N Carpp, Ludwig Geistlinger, Federico Martini, Kevin Rue-Albrecht, Davide Risso, Charlotte Soneson, Levi Waldron, Hervé Pagès, Mike L Smith, Wolfgang Huber, Martin Morgan, Raphael Gottardo, and Stephanie C Hicks. Orchestrating single-cell analysis with bioconductor. *Nat. Methods*, pages 1–9, December 2019.

Yongzheng Cong, Yiran Liang, Khatereh Motamedchaboki, Romain Huguet, Thy Truong, Rui Zhao, Yufeng Shen, Daniel Lopez-Ferrer, Ying Zhu, and Ryan T Kelly. Improved single cell proteome coverage using Narrow-Bore packed NanoLC columns and ultrasensitive mass spectrometry. *Anal. Chem.*, January 2020.

Laurent Gatto. *QFeatures: Quantitative features for mass spectrometry data*, 2020. URL `https://github.com/RforMassSpectrometry/QFeatures`. R package version 0.7.0.

Aaron Lun and Davide Risso. *SingleCellExperiment: S4 Classes for Single Cell Data*, 2020. R package version 1.10.1.

Harrison Specht, Edward Emmott, Toni Koller, and Nikolai Slavov. High-throughput single-cell proteomics quantifies the emergence of macrophage heterogeneity. June 2019.

Sarah M Williams, Andrey V Liyu, Chia-Feng Tsai, Ronald J Moore, Daniel J Orton, William B Chrisler, Matthew J Gaffrey, Tao Liu, Richard D Smith, Ryan T Kelly, Ljiljana Paša-Tolić, and Ying Zhu. Automated coupling of nanodroplet sample preparation with liquid Chromatography-Mass spectrometry for High-Throughput Single-Cell proteomics. *Anal. Chem.*, July 2020.