# Replicate Specht et al. 2019 mass spectrometry-based single-cell proteomics analysis

Laurent Gatto, Christophe Vanderaa

CBIO, de Duve Institute, UCLouvain

18 August 2020

# Outline

# State of the art in MS-SCP



MS-SCP: Mass spectrometry-based single-cell proteomics
MS-SCP consist of shotgun proteomics at single-cell level

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

**BUT**

Lack of standardized analysis software

# Our objectives



Provide a suite of software package dedicated to MS-SCP that fulfill:

- User-friendly
- Computationaly efficient
- Modularity: integrate other software packages
- Promote reproducibility
- Platform-independent
- Free of charge

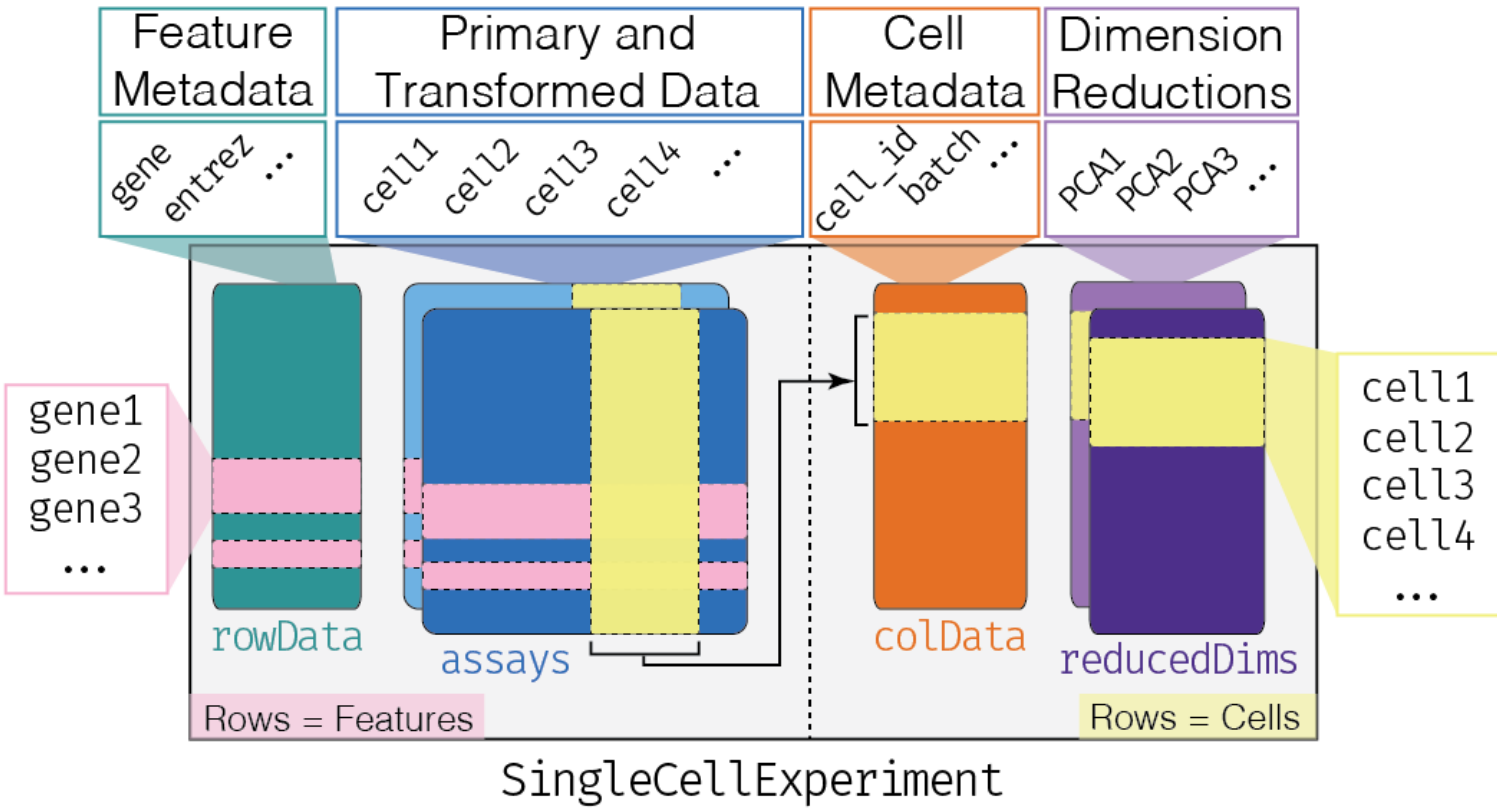
R/Bioconductor is an ideal environment

# Outline

# SingleCellExperiment



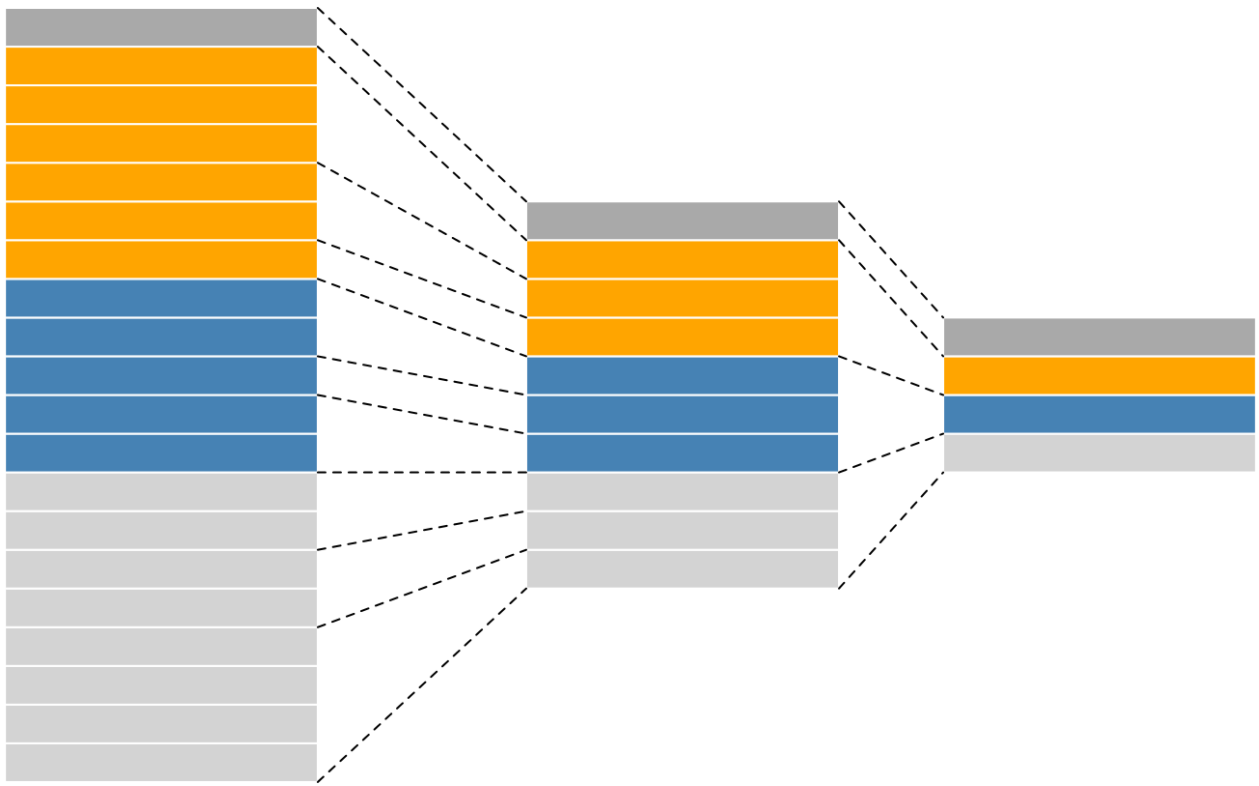
SingleCellExperiment: provides dedicated framework for single-cell data analysis.
Available on Bioconductor.



# QFeatures



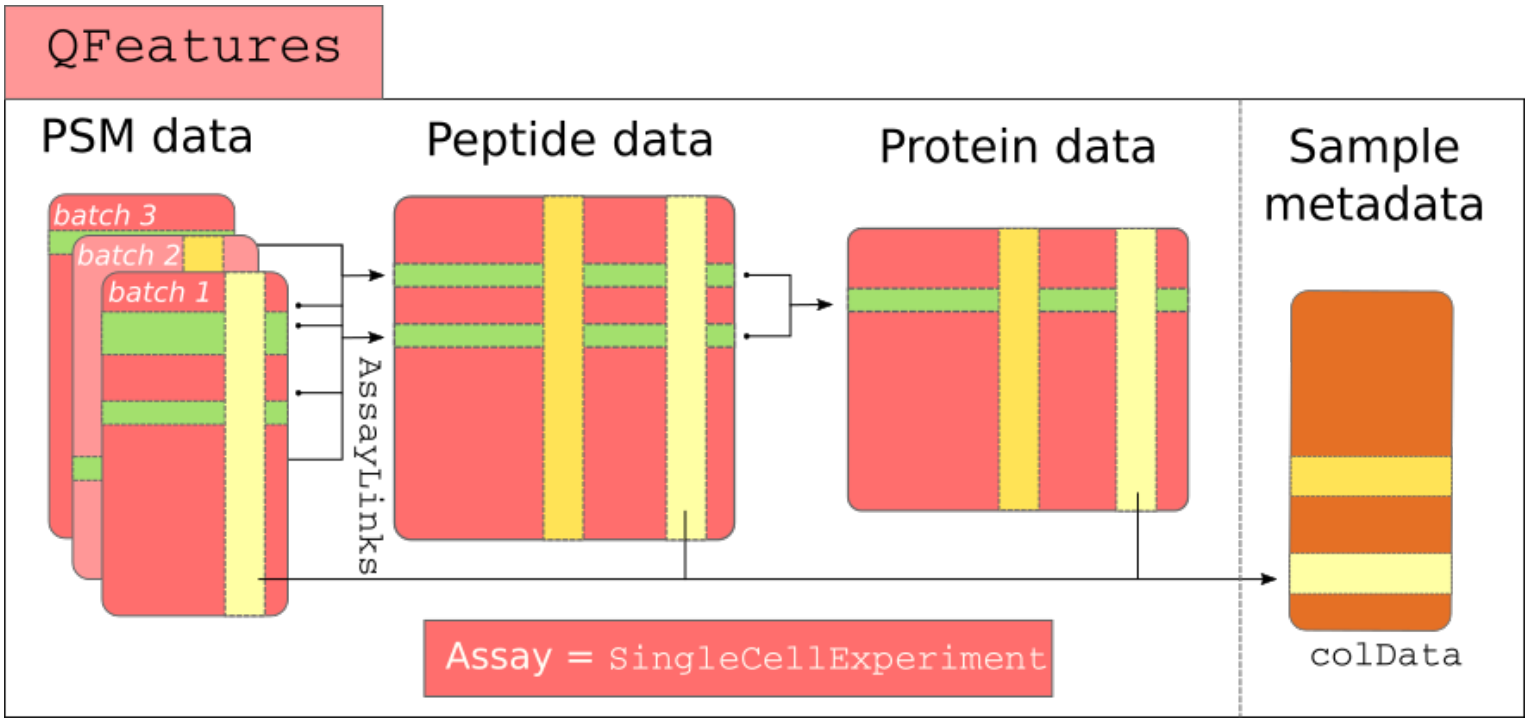
QFeatures: data framework dedicated to manipulate and process MS-based quantitative data.
Submitted to Bioconductor.



# Data framework



MS-SCP data framework = SingleCellExperiment + QFeatures



# MS-SCP package suite



scpdata: distributes published MS-SCP datasets (e.g. SCoPE2 dataset)
scp: provides functionality for manipulating the MS-SCP data structure

# Outline

# Load data



Load the SCoPE2 dataset called `specht2019v2`

```
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
```

# Feature selection



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[1]: expected rate of wrongly assigned features to a given peptide
- Cell median CV[2]: reliability of the protein quantification summarized over each cell.

Example:

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

Source code in `scp`

[1] false discovery rate

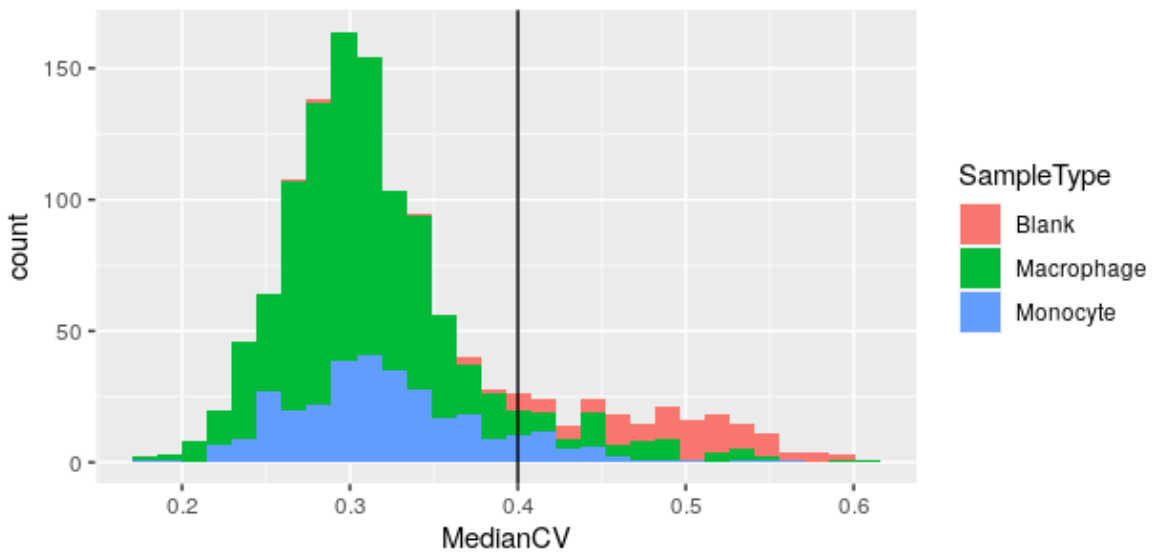
[2] coefficient of variation

# 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 to correct

```
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

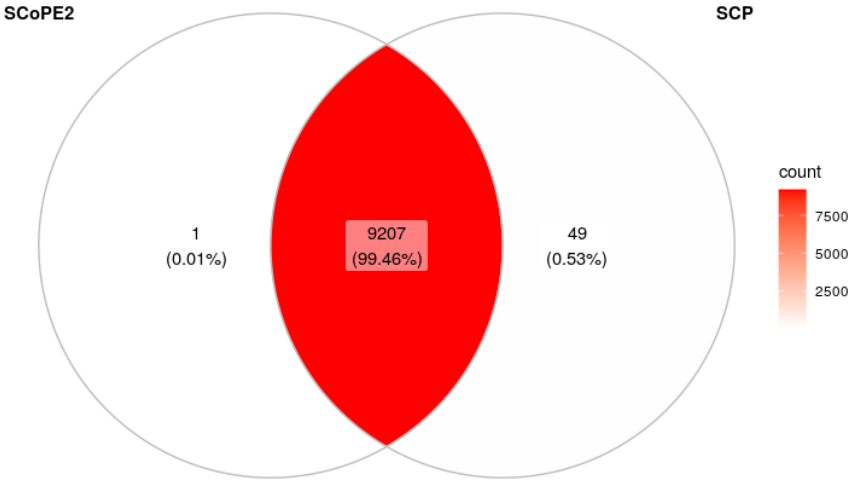
```
addAssay(specht2019v2,
         sce,
         name = "proteins_batchC") %>%
addAssayLinkOneToOne(from = "proteins",
                     to = "proteins_batchC")
```

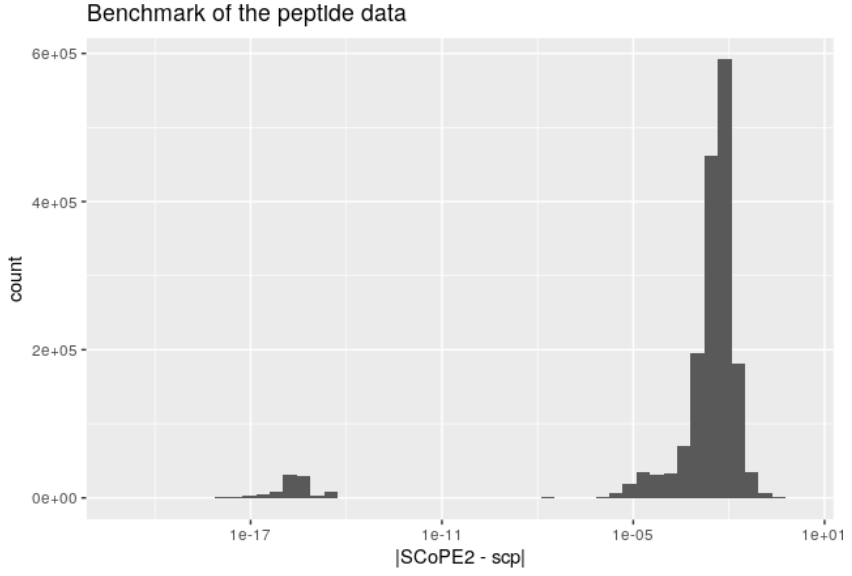
# Outline

# Benchmarking data matrices

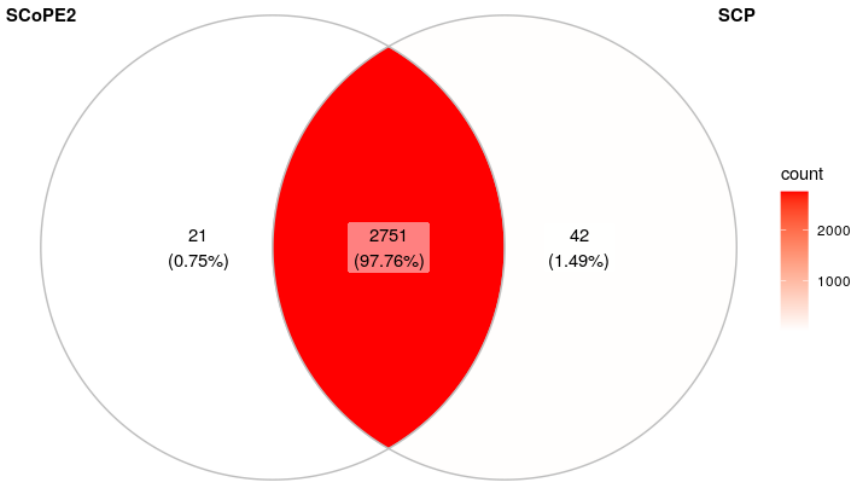


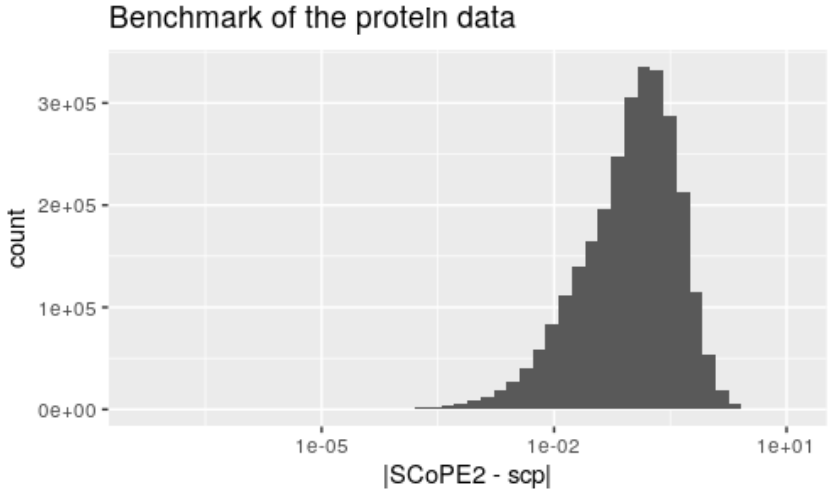
## Peptides




## Proteins



# Replicate figures from SCoPE2 (1)

# Replicate figures from SCoPE2 (2)

# Replicate figures from SCoPE2 (3)

# Outline

# Conclusion



- `scp` package suite provides a standardized environment for performing MS-SCP data analysis
- Flexibly reproduce existing analyses from different groups or protocoles (multiplex vs label free)

Advantages:

- Allow automation of the analysis
- Facilitate new computational developments
- Promotes reproducibility
- Increases field visibility

# Resources

## Packages

- `scp`: GitHub repository `UClouvain-CBIO/scp`
- `scpdata`: GitHub repository `UClouvain-CBIO/scpdata`
- `QFeatures`: GitHub repository `rformassspectrometry/QFeatures`
- `SingleCellExperiment`: Bioconductor

## SCoPE2 reproduction vignette

Available at...

## Slides and source code

Available at...

# Acknowledgements