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
Belgium

18 August 2020

Outline

Introduction

scp package

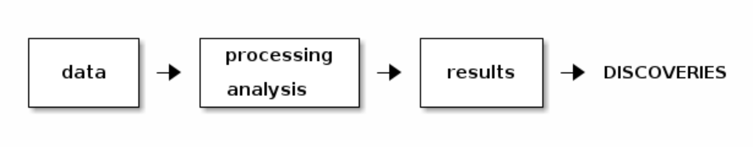
scp showcase

Replication results

Conclusion

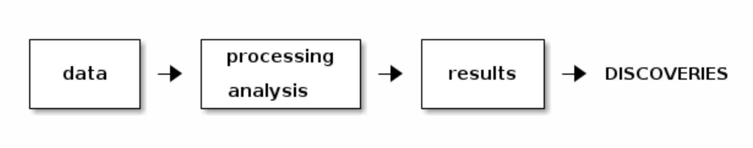
The value of replication (1)

- Expectation

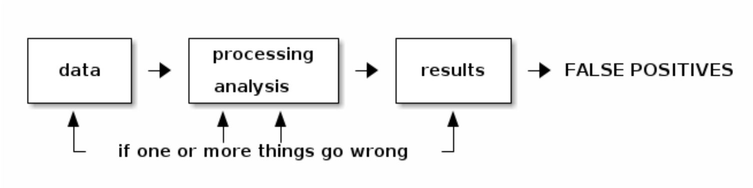


The value of replication (1)

- Expectation



- Reality



The value of replication (2)

- **Replication-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 standardised and principled data and analysis that is broadly applicable.
- Open, transparent and reproducible computational infrastructure to further improve data analysis and interpretation.
- R and Bioconductor (**?**) offer an ideal environment to attain these goals.

Implemented in the `scp` package.

Outline

Introduction

scp package

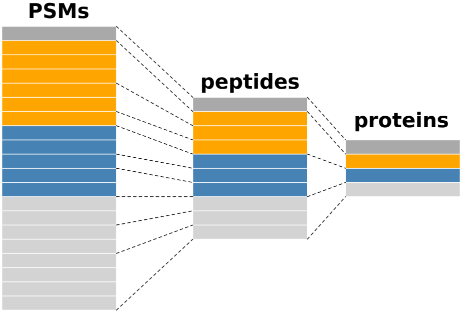
scp showcase

Replication results

Conclusion

Data infrastructure: QFeatures[1]

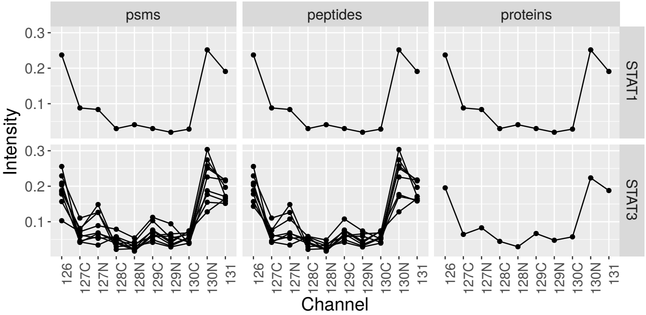
`QFeatures`: data framework dedicated to manipulate and process MS-based quantitative data.



[1] ?

Data infrastructure: QFeatures[1]

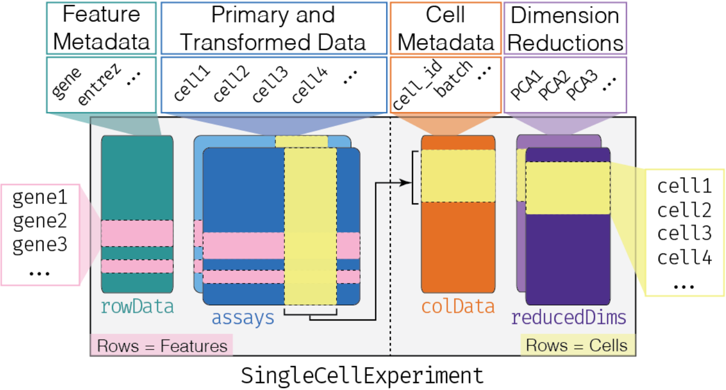
`QFeatures`: data framework dedicated to manipulate and process MS-based quantitative data.



[1] ?

Data infrastructure: SingleCellExperiment[2,3]

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

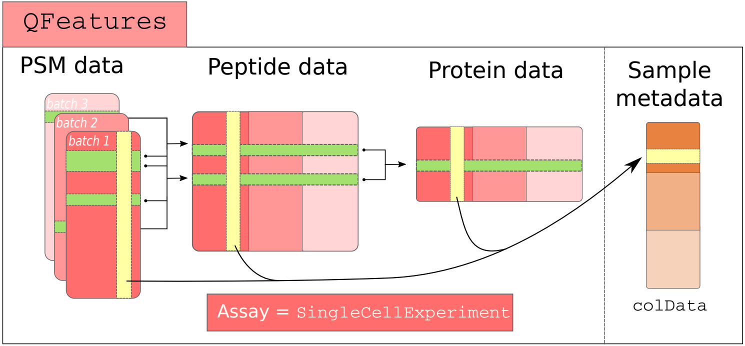


[2] ?

[3] ?

Data infrastructure: scp

`scp` = `SingleCellExperiment` + `QFeatures`



AND functions dedicated to processing and analyzing SCP data.

Load data

Load the SCoPE2 dataset called `specht2019v2`[4]

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

Dataset overview

```r
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 (generated by MaxQuant, ProteomeDiscoverer, ...) can be converted to `QFeatures` using the `readSCP()` function.

[4] ?

Metadata

Sample metadata common to **all assays** are stored in a single table, the `colData`

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 aggregation
5. Joining sets
6. Single cells filtering based on median CV
7. Normalization
8. Removal of highly missing peptides
9. 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 aggregation
5. Joining sets
6. Single cells filtering based on median CV
7. Normalization
8. Removal of highly missing peptides
9. 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

Protein 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 aggregation
5. Joining sets
6. **Single cells filtering based on median CV**
7. Normalization
8. **Removal of highly missing peptides**
9. **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**

Protein data

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

Outline

Introduction

scp package

scp showcase

Replication results

Conclusion

Data 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

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

Source code in `QFeatures`

Data filtering: compute QC metrics

Some QC metrics are not compued by MaxQuant:

- Sample to carrier ratio: discard samples with intensities higher than expected
- Peptide FDR: expected proportion of features wrongly assigned to a given peptide
- Cell median coefficient of variation: reliability of the protein quantification in a cell.

Example:

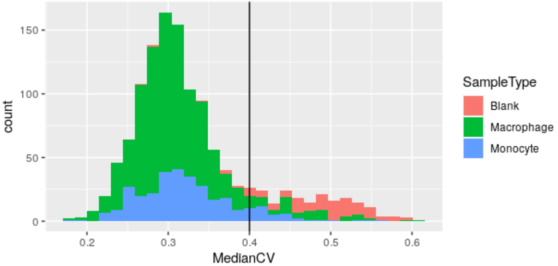
```
computeMedianCV(specht2019v2,
                i = "peptides",
                proteinCol = "protein",
                peptideCol = "peptide",
                batchCol = "Set")
```

Source code in `scp`

Data filtering: plot QC metrics

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

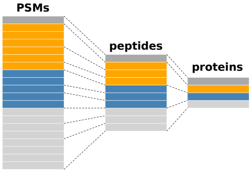
```r
library(tidyverse)
specht2019v2[["peptides"]] %>%
  colData %>%
  data.frame %>%
  ggplot(aes(x = MedianCV,
                fill = SampleType)) +
  geom_histogram() +
  geom_vline(xintercept = 0.4)
```



Feature aggregation

Feature aggregation = combine features into a higher-level structure.

Example: aggregate peptides to proteins



- Combine the quantitative data from multiple peptides to a single protein
- Store the relationship between the protein and the aggregated peptides

```
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")
```

Remove highly-missing features (e.g. $>=$ 99 %)

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

Impute missing data

```
impute(specht2019v2,
       i = "proteins",
       method = "knn",
       k = 3)
```

Source code in `QFeatures`

Data transformation

Common data transformation can easily be applied such as **log-transformation** or **normalization**.

Example: *log_2*-transformation:

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

Source code in `QFeatures`

Custom functions

Custom function can be applied to the data set, for example batch correction using ‘ComBat‘. Three-step procedure:

1. Extract the assay to process

```r
x <- specht2019v2[["proteins"]]
```

2. Apply the custom function

```r
assay(x) <- ComBat(assay(x), ...)
```

3. Add the processed assay in the dataset

```r
addAssay(specht2019v2, x, name = "proteins_batch_corrected")
```

Outline

Introduction

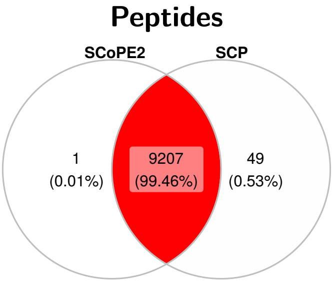
scp package

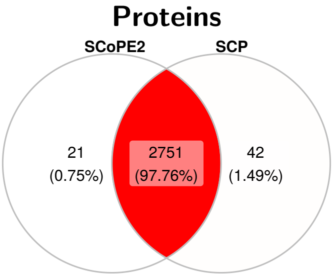
scp showcase

Replication results

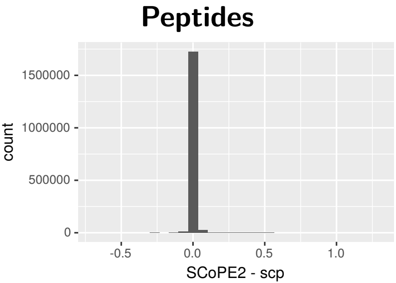
Conclusion

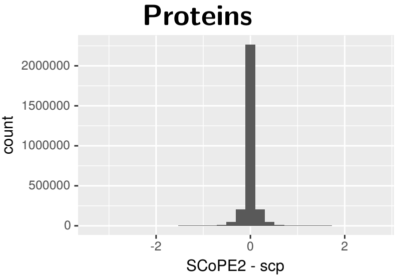
Selected features





Numerical comparison

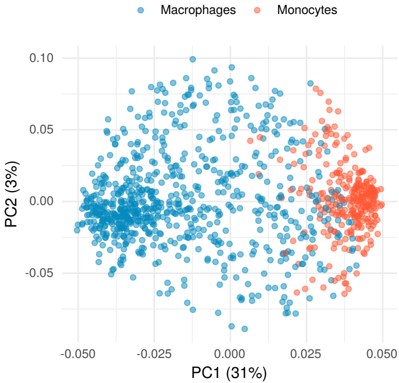




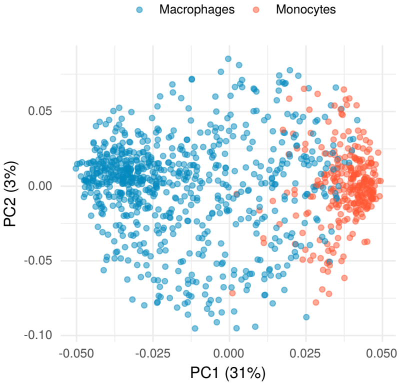
Replicate weighted PCA

Weighted PCA on the **protein** data

SCoPE2 (Figure 3a in preprint)



scp

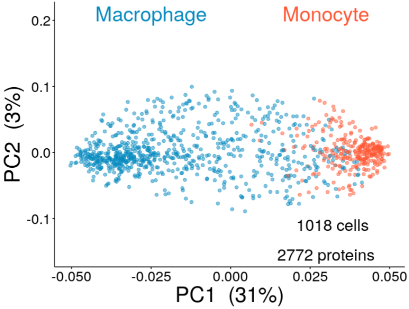


Note about reproducibility

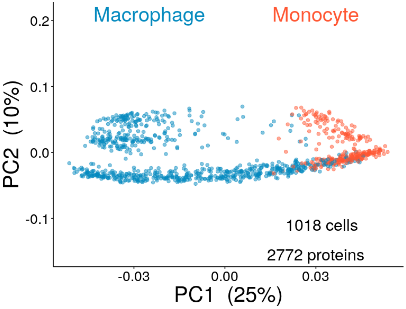
Good software development includes continuous maintenance and improvement **BUT** might impact reproducibility

Example: batch correction using 2 versions of the `ComBat` algorithm (`sva` package)

`sva 3.34.0`



`sva 3.36.0`



Documenting software version is **essential** for reproducible work

Outline

Introduction

scp package

scp showcase

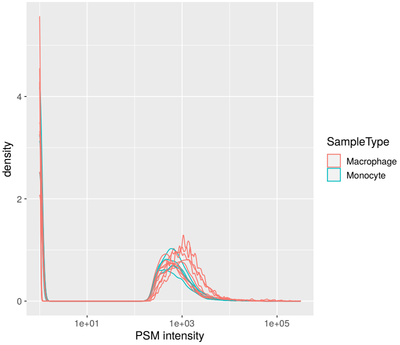
Replication results

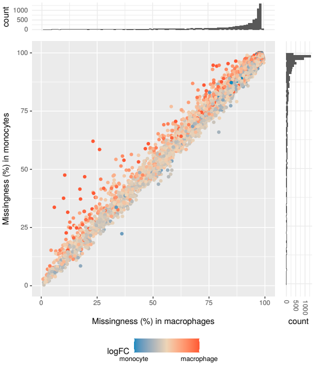
Conclusion

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 (**??**).
- `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