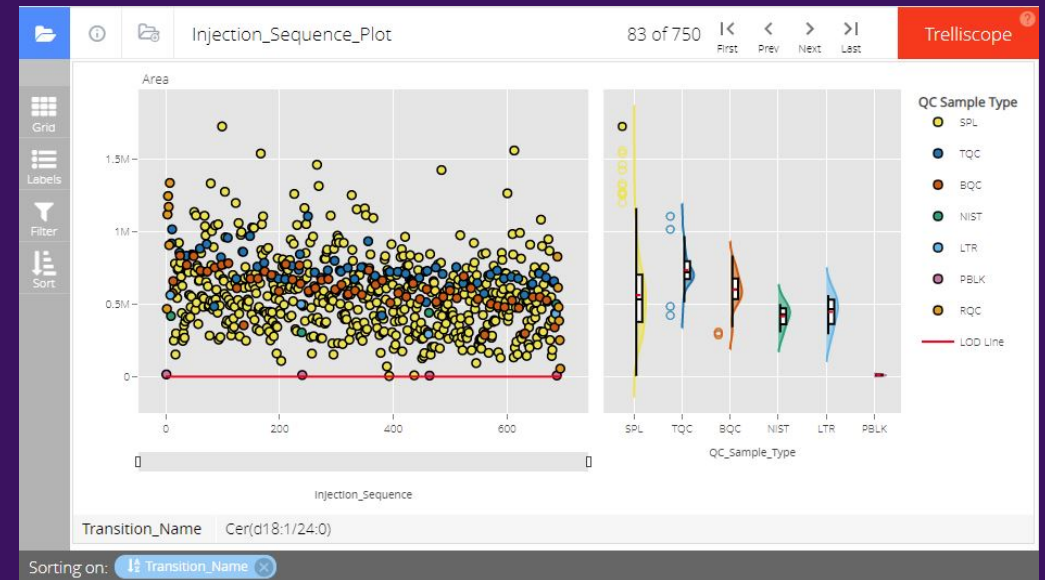


Viewing Multiple Interactive Plots Using `plotly` and `trelliscopejs`

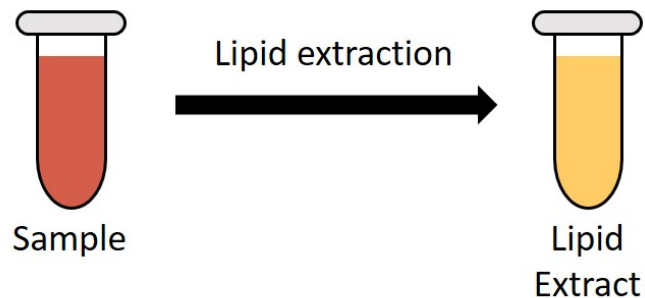


R/Medicine 2022



Jeremy Selva    

How Lipids are measured (Targeted Lipidomics)



Lipid measurement

Transition_Name	Precursor_Ion	Product_Ion
Cer d18:1/12:0 (ISTD)	482.4	264.4

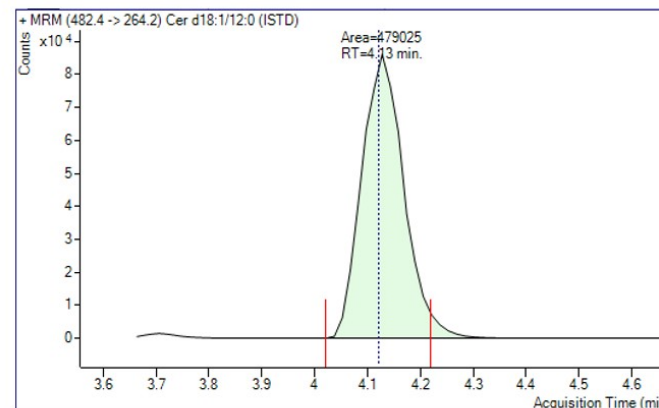


Liquid Chromatography/Mass Spectrometry

Sample_Name	Cer d18:1/12:0 (ISTD)
Sample 01	479025
Sample 02	570275
Sample 03	374854
Sample 04	319491
Sample 05	231691

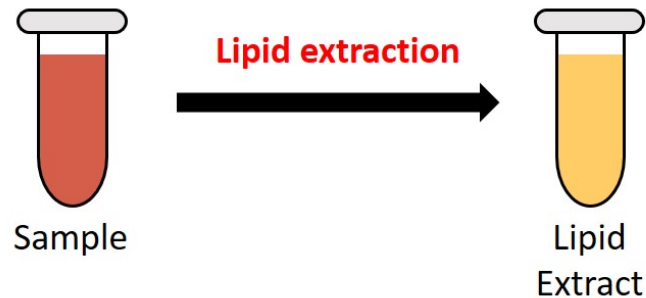
Peak Area Data

Peak integration



Source of "Unwanted" Variation

- Lipid Extraction
 - Different pipettes used to do extraction.
- Lipid Measurement
 - Contaminant ions



Lipid measurement

Transition_Name	Precursor_Ion	Product_Ion
Cer d18:1/12:0 (ISTD)	482.4	264.4

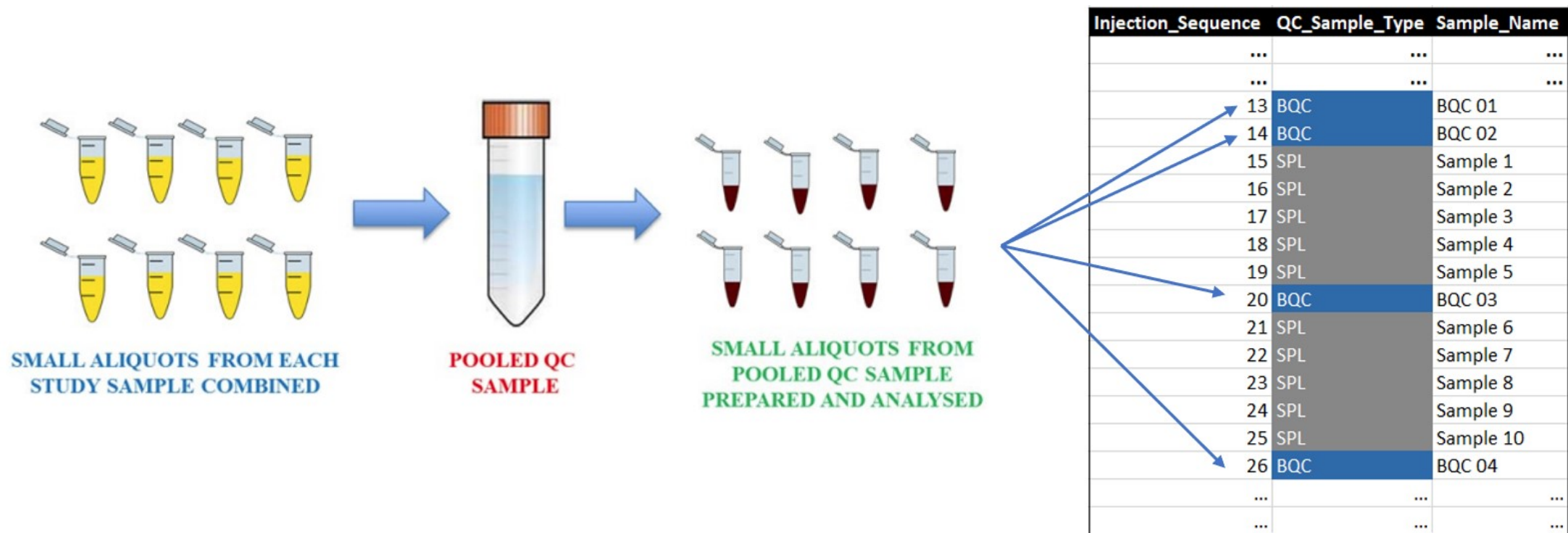


Liquid Chromatography/Mass Spectrometry

Quality Control (QC) Samples

Quality control samples helps to check the severity of unwanted variation.

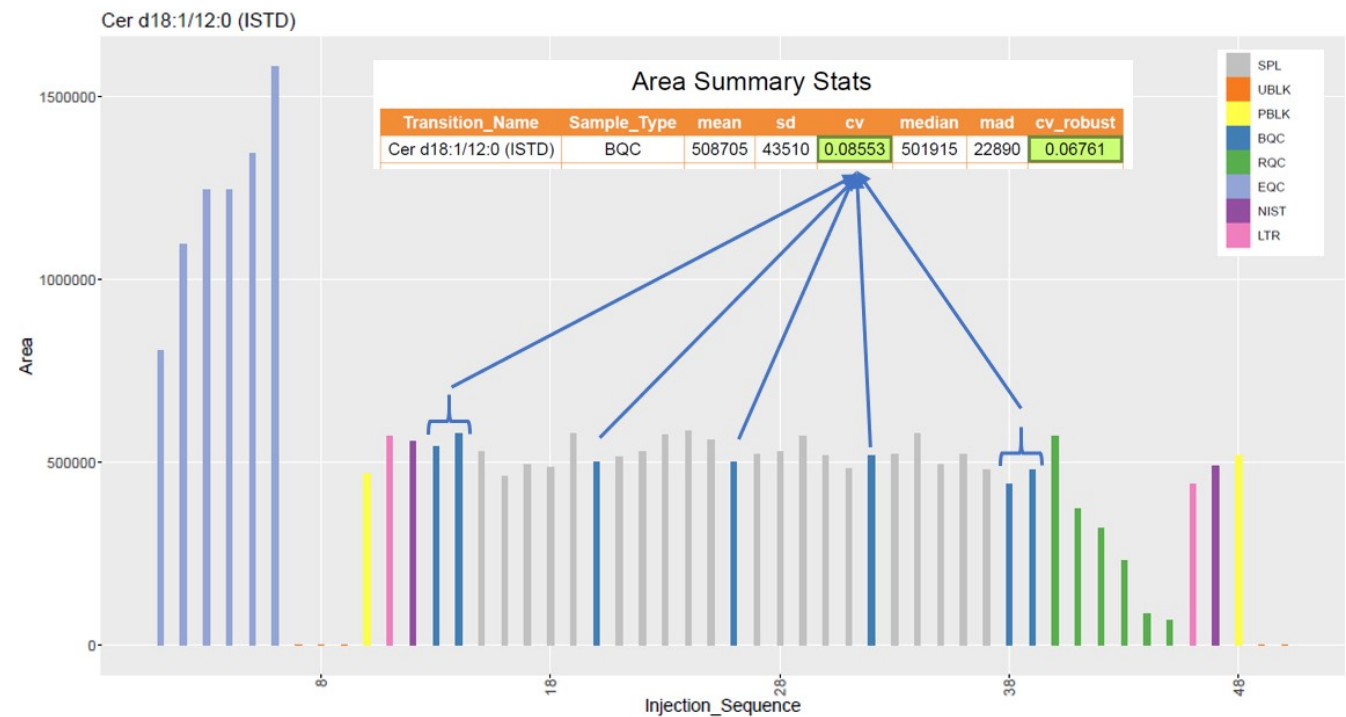
One example is the Batch Quality Control (BQC) to measure variation during lipid extraction and measurement.



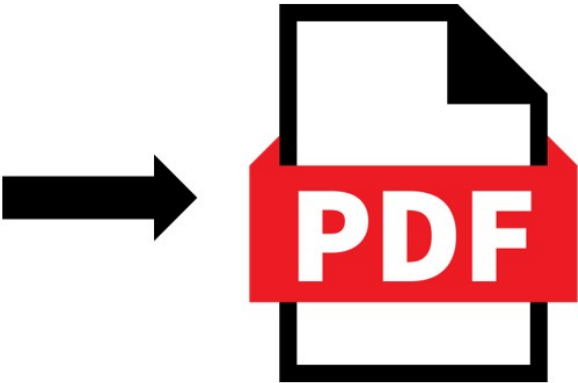
Quality Control Plot

An injection sequence bar chart with some statistical summary is created for each transition. Keep transitions with high BQC coefficient of variation (CV).

Report results as a pdf file of injection sequence bar charts. (One transition per page)



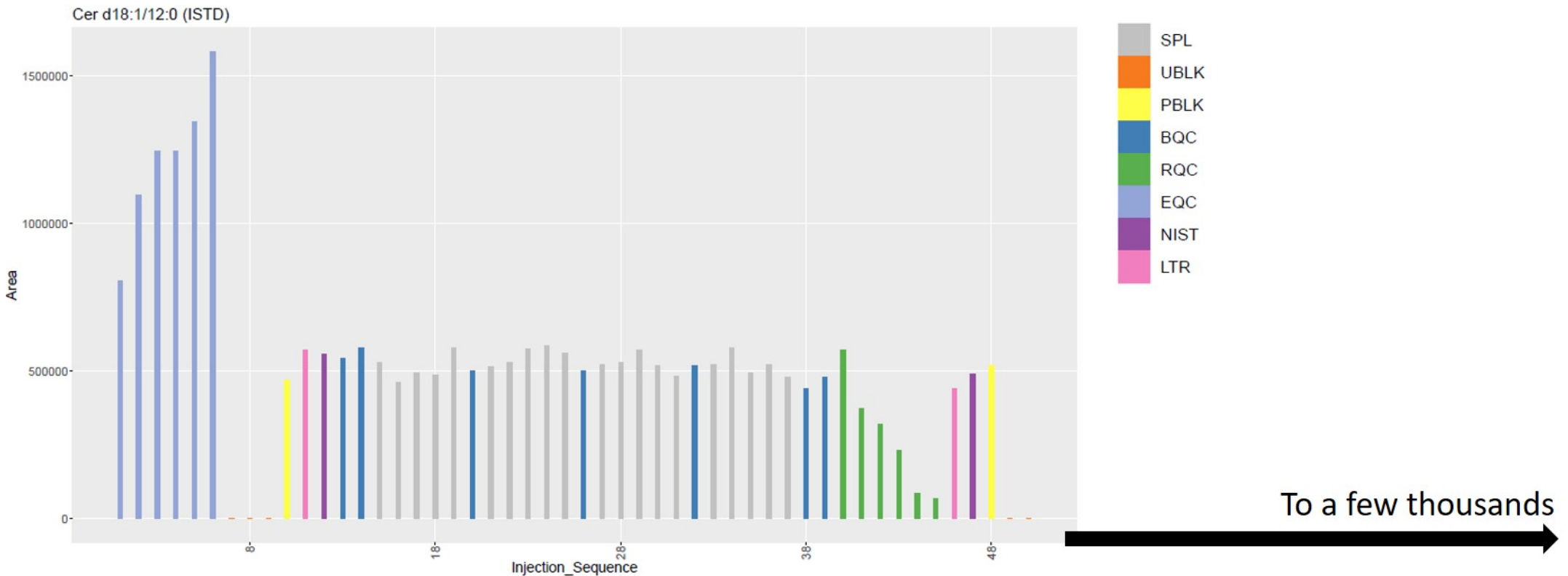
Injection Sequence Plot for Cer d18:1/12:0 (ISTD)



Motivation for using **plotly** and **trelliscopejs**

Number of samples are getting larger (increase to few thousands).

Static bar charts and scatter plots will not work.

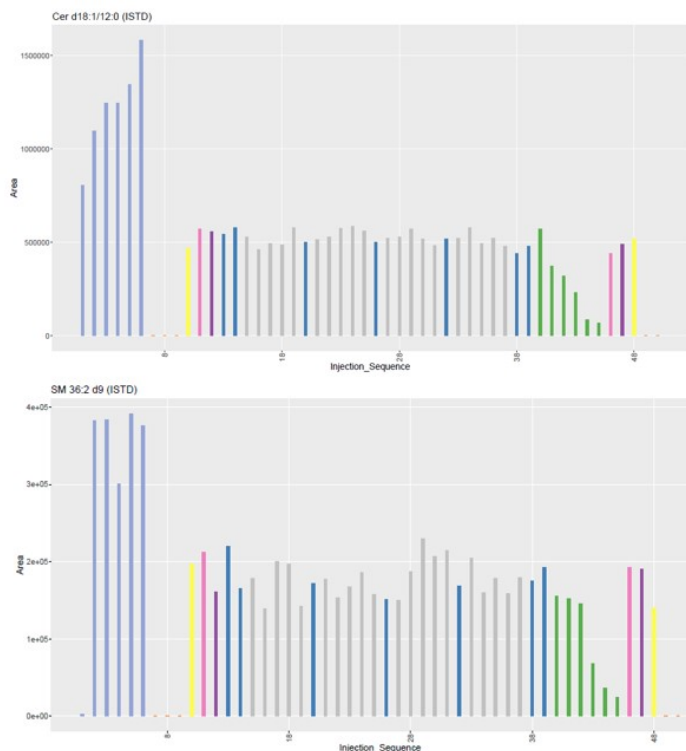


Motivation for using **plotly** and **trelliscopejs**

Number of transitions have increased to over 500. This give a > 500 -pages pdf file full of plots.

Transition_Name	Precursor_Ion	Product_Ion
...
...
Cer d18:1/12:0 (ISTD)	482.4	264.4
LPC 18:1 d7 (ISTD)	529.7	184.1
SM 36:2 d9 (ISTD)	739.2	184.1
...
...

> 500 transition list



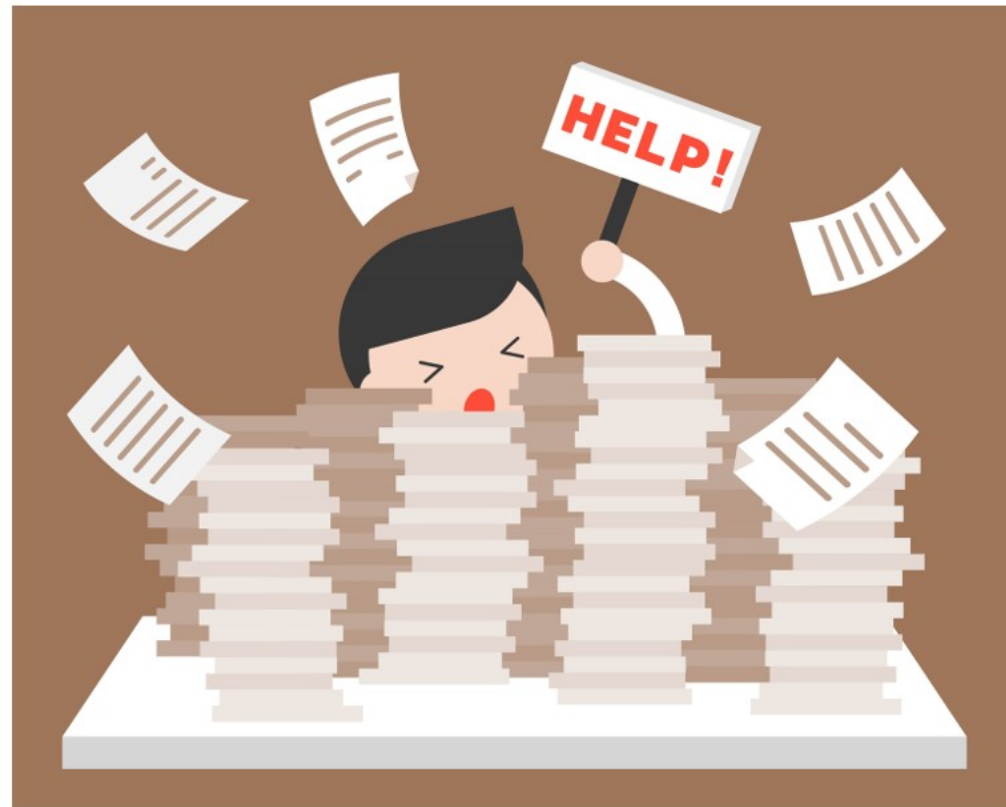
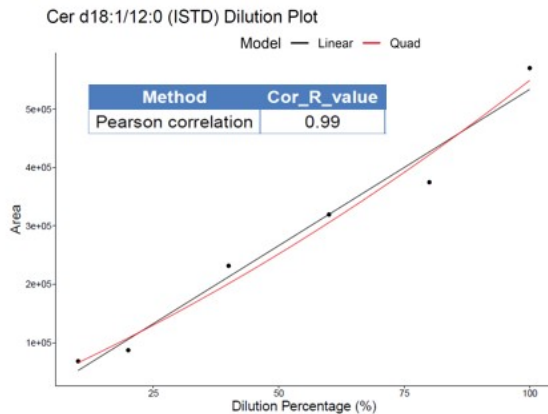
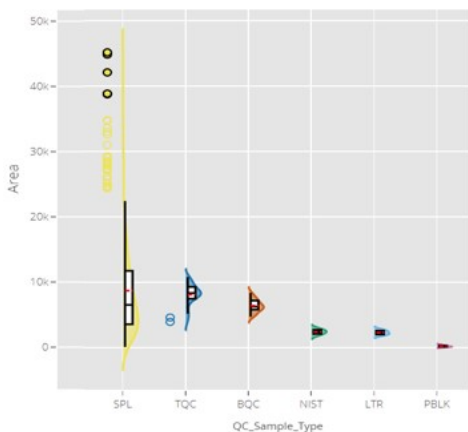
SPL
UBLK
PBLK
BQC
RQC
EQC
NIST
LTR

> 500 plots

Motivation for using **plotly** and **trelliscopejs**

Different QC sample types give rise to different plot type, resulting in **multiple** > 500-pages pdf files.

QC_Sample_Type	Name	Description
SPL	Study Sample	"Unknown" sample
BQC	Batch QC	Pooled samples to check variability during lipid extraction and measurement from instrument
TQC	Technical QC	Pooled lipid extracts to check variability during lipid measurement from instrument
RQC	Response QC	Dilution series to check for linear response/saturation (e.g. TQC)
LTR	Long Term Reference Plasma with ISTD	Long Term Reference Plasma with ISTD
NIST	NIST SRM1950 Reference Plasma with ISTD	NIST SRM1950 Reference Plasma with ISTD
PBLK	Processed Blank	Processed or Extracted solvent with ISTD



Select a Display to Open

no thumbnail

no thumbnail

Dilution_Plot

750 panels, 2022-08-02 17:18

Injection_Sequence_Plot

750 panels, 2022-08-02 17:28

CLOSE

Quarto Example

Thanks to open science, I am able to create a walk through example using Quarto  .

Article | [Open Access](#) | [Published: 10 January 2022](#)

Lipidomic profiling of human serum enables detection of pancreatic cancer

[Denise Wolrab](#), [Robert Jirásko](#), ... [Michal Holčápek](#)  [+ Show authors](#)

[Nature Communications](#) **13**, Article number: 124 (2022) | [Cite this article](#)

9421 Accesses | 5 Citations | 41 Altmetric | [Metrics](#)

Data availability

All data necessary to support the conclusions are available in the manuscript or supplementary information. Source data are provided with this paper. Raw data, instructions for software handling, and the software are deposited at figshare.com:<https://figshare.com/s/cc087785ca362af7118e>—(UHPSFC/MS: Phase I and Phase II), <https://figshare.com/s/e336bdf3a52f04c2de1f>—(Shoutgun-MS (LR and HR): Phase II), <https://figshare.com/s/cb071be45cd91a7c90e2>—(MALDI-MS: Phase I), <https://figshare.com/s/1fd10f273b049b93fa24>—(RP-UHPLC/MS: Phase II). [Source data](#) are provided with this paper.

Supplementary information

[Supplementary Information](#)

[Peer Review File](#)

[Description of Additional Supplementary Files](#)

[Supplementary Data 1-20](#)

[Reporting Summary](#)

Table of contents

- [Introduction](#)
- [R Packages Used](#)
- [Input Data](#)
- [Nested Data](#)
- [Annotate Transition Names](#)
- [Calculate Transition Statistics](#)
- [Plotting Palette](#)
- [Plot Dilution Curves](#)
- [Plot Injection Sequence](#)
- [Cognostics](#)
- [Create Trellis Plot](#)
- [Package References](#)
- [References](#)

Quarto Report Example With Plotly and Trelliscopejs

AUTHOR

Jeremy Selva   

 Code 

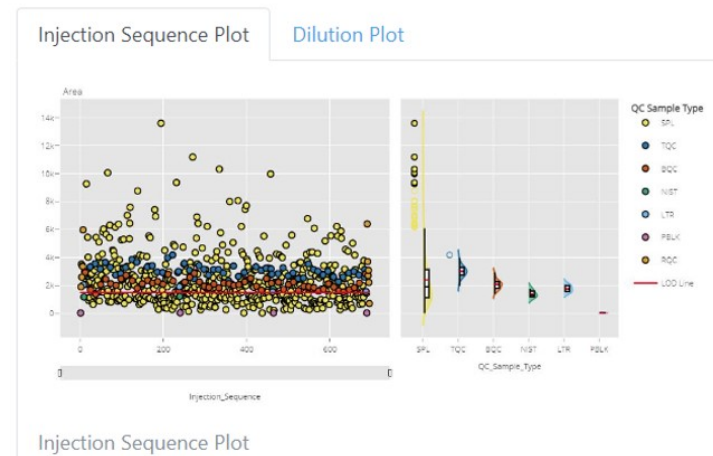
Show All Code

Hide All Code

View Source

Introduction

Here is a report showing how to create injection sequence plot and dilution plot using [plotly](#) for each transition.



Vector Of Plots

Use `dplyr::mutate` and `trelliscopejs::pmap_plot` to save `plotly` plots as a new column.

```
dilution_plot_table <- summary_data |>
dplyr::mutate(
  dilution_panel = trelliscopejs::pmap_plot(
    list(dilution_data = .data[["dilution_data"]]),
    plot_dil_curve_plotly,
    dilution_title = "",
    dilution_pal = dilution_pal,
    sample_name_var = "Sample_Name",
    dil_batch_var = "Dilution_Batch_Name",
    conc_var = "Relative_Sample_Amount[%]",
    conc_var_units = "%",
    conc_var_interval = 25,
    signal_var = "Area",
    x_axis_title = "Relative Sample Amount",
    y_axis_title = "Area",
    y_axis_rangemode = "normal",
    lin_reg_name = "lin reg",
    quad_reg_name = "quad reg",
    show_legend = TRUE,
    show_y_axis_title = TRUE,
    annotation_var = c("Dilution_Batch_Name",
                      "Injection_Sequence",
                      "Acquisition_Time_Stamp",
                      "Vial_Position")
  )
) |>
dplyr::select(-c("dilution_data", "sample_data"))
```

Map over multiple inputs simultaneously and return a vector of plots

Source: R/tidy.R

Map over multiple inputs simultaneously and return a vector of plots

`map2_plot(.x, .y, .f, ...)`

`pmap_plot(.l, .f, ...)`

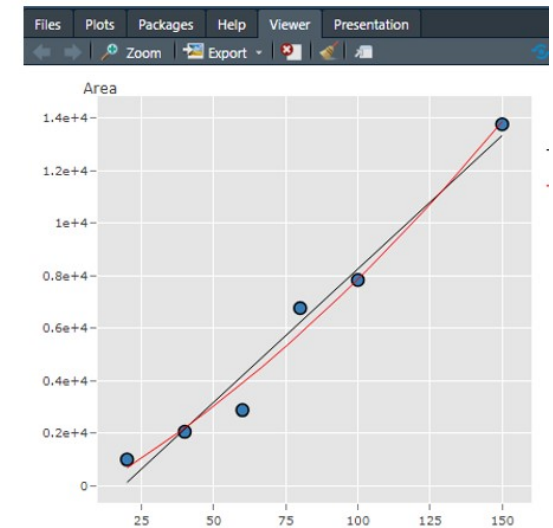
Trelliscopejs_In_Quarto_Example - main - RStudio

Transition_Name	dilution_panel	Lipid_Name
1 AcylCarnitine 10:0	<code>list(x = list(visdat = list("77644c483021" = funct [...]</code>	CAR 10:0
2 AcylCarnitine 10:1	<code>list(x = list(visdat = list("776423dd5818" = funct [...]</code>	CAR 10:1
3 AcylCarnitine 12:0	<code>list(x = list(visdat = list("776449445435" = funct [...]</code>	CAR 12:0
4 AcylCarnitine 12:1	<code>list(x = list(visdat = list("77647ac9148a" = funct [...]</code>	CAR 12:1
5 AcylCarnitine 13:0	<code>list(x = list(visdat = list("77642fcb50cb" = funct [...]</code>	CAR 13:0
6 AcylCarnitine 14:0	<code>list(x = list(visdat = list("7764393e52d3" = funct [...]</code>	CAR 14:0

Showing 1 to 6 of 750 entries, 44 total columns

Console Background Jobs

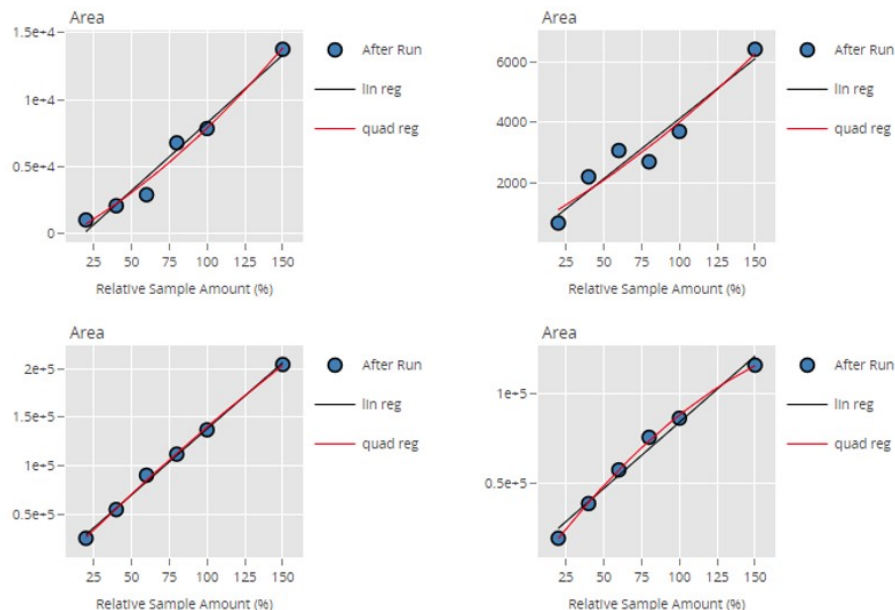
```
R 4.2.1 · D:/Trelliscopejs_In_Quarto_Example/
> dilution_plot_table$dilution_panel[[1]]
>
```



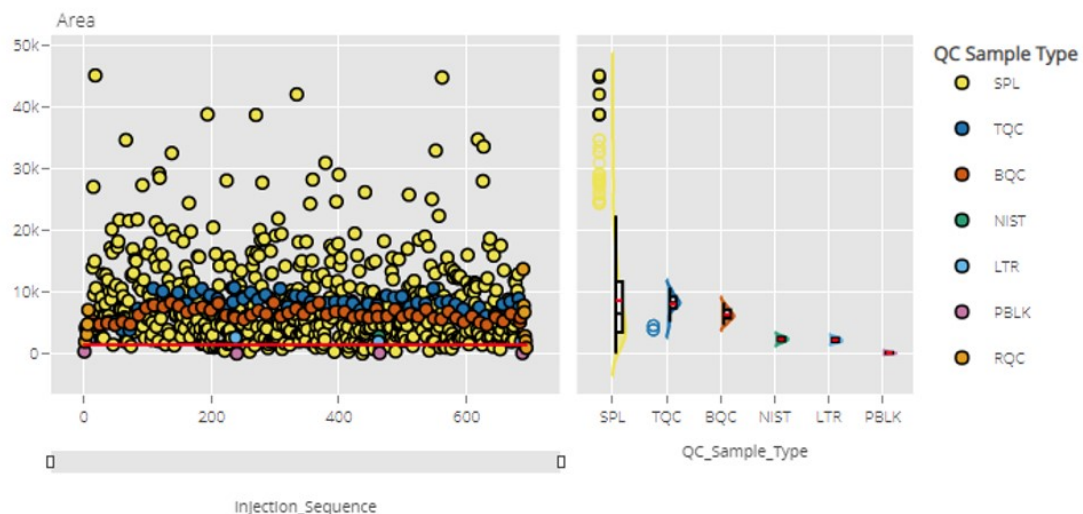
Patching **plotly** Plots

Use `manipulateWidget::combineWidgets` and `plotly::subplot` to patch different **plotly** plots.

```
manipulateWidget::combineWidgets(ncol = 2,  
  nrow = 2,  
  dilution_plot_table$dilution_panel[[1]],  
  dilution_plot_table$dilution_panel[[2]],  
  dilution_plot_table$dilution_panel[[3]],  
  dilution_plot_table$dilution_panel[[4]])
```

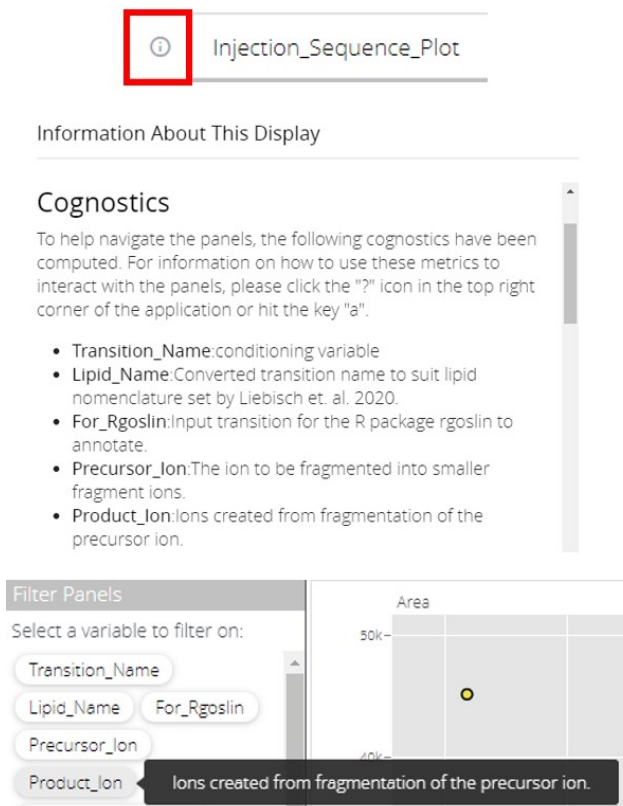


```
combined_plots <- plotly::subplot(scatterplot_panel,  
  raincloud_panel,  
  widths = c(0.6, 0.4),  
  margin = 0.01,  
  shareX = TRUE,  
  shareY = TRUE  
)
```



trelliscopejs cognostics

Providing metadata information as **trelliscopejs** cognostics can help to improve the user experience. Refer to the Quarto   example on how to do it.



Injection_Sequence_Plot

Information About This Display

Cognostics

To help navigate the panels, the following cognostics have been computed. For information on how to use these metrics to interact with the panels, please click the "?" icon in the top right corner of the application or hit the key "a".

- **Transition_Name**:conditioning variable
- **Lipid_Name**:Converted transition name to suit lipid nomenclature set by Liebisch et. al. 2020.
- **For_Rgoslin**:Input transition for the R package rgoslin to annotate.
- **Precursor_Ion**:The ion to be fragmented into smaller fragment ions.
- **Product_Ion**:Ions created from fragmentation of the precursor ion.

Filter Panels

Select a variable to filter on:

Transition_Name

Lipid_Name For_Rgoslin

Precursor_Ion

Product_Ion Ions created from fragmentation of the precursor ion.

Table of contents

- Introduction
- R Packages Used
- Input Data
- Nested Data
- Annotate Transition Names
- Calculate Transition Statistics
- Plotting Palette
- Plot Dilution Curves
- Plot Injection Sequence
- Cognostics**
 - Cognostics Conversion Functions
 - Cognostics Setup
 - Cognostics Creation
- Create Trellis Plot
- Package References
- References

Cognostics

To output these interactive plots as a trellis plot in html, we need to convert our current tibble and `dilution_plot_table` and `injection_sequence_table` into a cognostics dataframe.

The cognostics dataframe consist of three components

1. A group of columns known as conditioning variables. They will form the unique id of the trellis plot. As such, each row of these columns must be unique. In our example, the column used is `Transition_Name`
2. One column that holds the images to display each plot in the trellis. This column is known as the panel variable. We just created this column using `trelliscopejs::pmap_plot` earlier.
3. The other columns will be grouped as general cognostics columns.

Cognostics Conversion Functions

The relevant functions to convert all columns, except the panel variable, to cognostics objects are as follows.

► Code

Cognostics Setup

The function `trelliscopejs::cog` is used to convert a column into cognostic object. The following meta information must be provided in order for the conversion to be successful.

Results Distribution

To export a **trelliscopejs** object.

```
Console Background Jobs x
R 4.2.1 · D:/Trelliscopejs_In_Quarto_Example/
> getwd()
[1] "D:/Trelliscopejs_In_Quarto_Example"
```

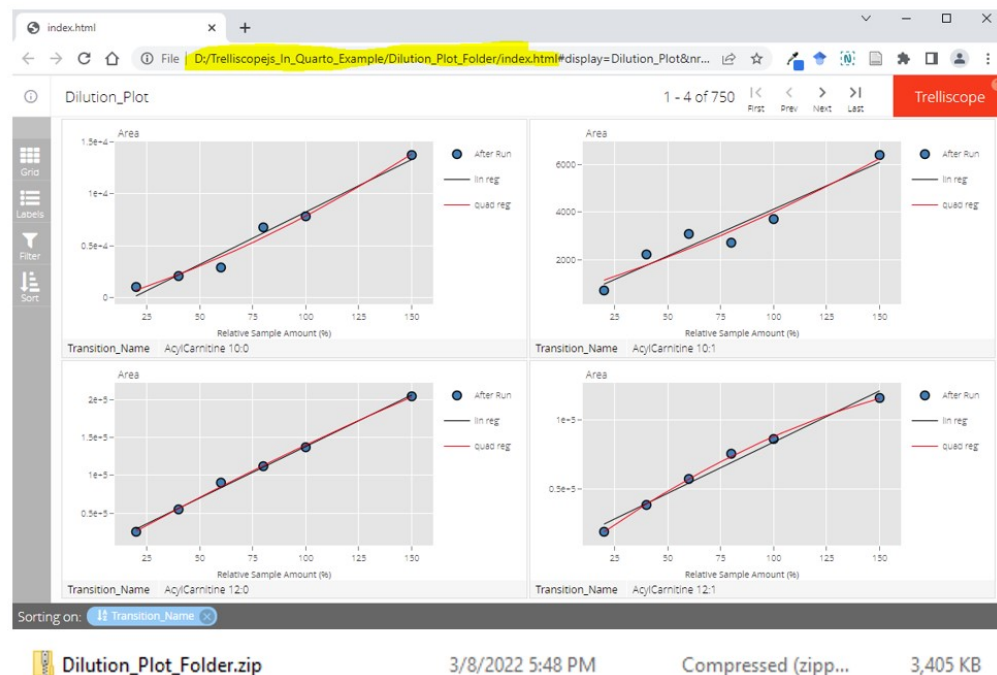
Call script via console

```
trelliscope_name <- "Dilution Plot"

dilution_trellis |>
  trelliscopejs::trelliscope(
    name = trelliscope_name,
    panel_col = "dilution_panel",
    path = "Dilution_Plot_Folder",
    state = list(sort = list(trelliscope
      nrow = 2,
      ncol = 2,
      height = 520,
      width = 1100,
      self_contained = FALSE,
      thumb = FALSE,
      auto_cog = FALSE
    )
  )
```

Data (D:) > Trelliscopejs_In_Quarto_Example > Dilution_Plot_Folder

Name	Date modified	Type	Size
appfiles	1/8/2022 7:27 PM	File folder	
lib	2/8/2022 5:01 PM	File folder	
index.html	2/8/2022 5:01 PM	Chrome HTML Do...	1 KB



Results Distribution

To export a Quarto document with `trelliscopejs` object.

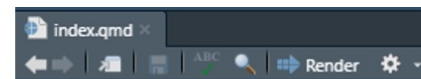
```
Console Background Jobs x
R 4.2.1 · D:/Trelliscopejs_In_Quarto_Example/ ↗
> getwd()
[1] "D:/Trelliscopejs_In_Quarto_Example"
```

```
trelliscope_name <- "Dilution Plot"

dilation_trellis |>
  trelliscopejs::trelliscope(
    name = trelliscope_name,
    panel_col = "dilation_panel",
    path = "Dilution_Plot_Folder",
    state = list(sort = list(trelliscope
nrow = 2,
ncol = 2,
height = 520,
width = 1100,
self_contained = FALSE,
thumb = FALSE,
auto_cog = FALSE
  )
```

Source Code

```
15 tidyverse-quiet: TRUE
16 format:
17   html:
18     theme:
19       light: cerulean
20       dark: cyborg
21   toc: true
22   toc-depth: 3
23   toc-location: left
24   number-sections: false
25   code-fold: show
26   code-overflow: scroll
27   code-line-numbers: true
28   code-copy: hover
29   code-tools: true
30   self-contained: false
31   smooth-scroll: true
```



Data (D:) > Trelliscopejs_In_Quarto_Example

Name	Date modified
^	
_extensions	3/8/2022 5:04 PM
data	3/8/2022 1:11 PM
Dilution_Plot_Folder	1/8/2022 7:27 PM
images	3/8/2022 11:11 AM
index_files	3/8/2022 5:31 PM
Injection_Sequence_Plot_Folder	2/8/2022 5:12 PM
renv	22/7/2022 7:13 PM
utils	31/7/2022 5:15 PM
.gitattributes	3/8/2022 4:09 PM
.gitignore	3/8/2022 4:05 PM
.Rhistory	3/8/2022 12:31 PM
.Rprofile	22/7/2022 7:11 PM
index.html	3/8/2022 5:31 PM
index.qmd	3/8/2022 5:24 PM
README.md	2/8/2022 4:30 PM
README.Rmd	2/8/2022 4:21 PM
renv.lock	2/8/2022 4:52 PM
Trelliscopejs_In_Quarto_Example.Rproj	3/8/2022 12:31 PM

Quarto_Example.zip

3/8/2022 6:59 PM

Results Distribution

Exported examples are found in the below [GitHub page](#).

JauntyJJS / Trelliscopejs_In_Quarto_Example Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

JauntyJJS add icons ✓ 93f28a2 1 hour ago 19 commits

Dilution_Plot_Folder	add icons	1 hour ago
Injection_Sequence_Plot_Folder	add icons	1 hour ago
_extensions	add icons	1 hour ago
data	Initial commit	yesterday
images	Add Dark Mode	6 hours ago
index_files	add icons	1 hour ago
utils	Add utils Folder	yesterday
.gitattributes	Use Git LFS to manage large zip files	3 hours ago
.gitignore	Remove some qmd files	3 hours ago
Dilution_Plot_Folder.zip	Use Git LFS to manage large zip files	3 hours ago
Injection_Sequence_Plot_Folder.zip	Use Git LFS to manage large zip files	3 hours ago
Quarto_Example.zip	Use Git LFS to manage large zip files	3 hours ago

About

Source code for a Quarto report using Plotly and Trelliscopejs

[jauntyjjs.github.io/trelliscopejs_in_quarto...](#)

Readme

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)


Packages

No packages published

[Publish your first package](#)

Enjoy The Conference

Summary

- Quality control samples are useful to check for unwanted variation in a targeted lipidomics workflow.
- [plotly](#) and [trelliscopejs](#) can help to explore many interactive plots in an effective way.
- Hope that the Quarto  example and advice provided are useful.

