



Viewing Multiple Interactive Plots with **plotly** And **trelliscopejs**

Jeremy Selva    

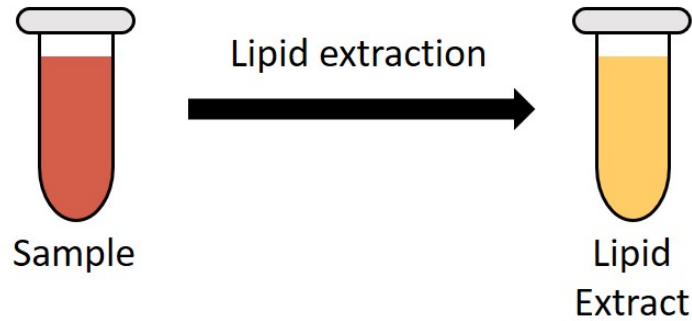
Introduction to Lipids

Lipids are organic compounds that are mostly insoluble in polar solvents like water.

The most common ones are cholesterol and triglycerides in blood lipid panel test.



How Lipids are measured (Targeted Lipidomics)



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

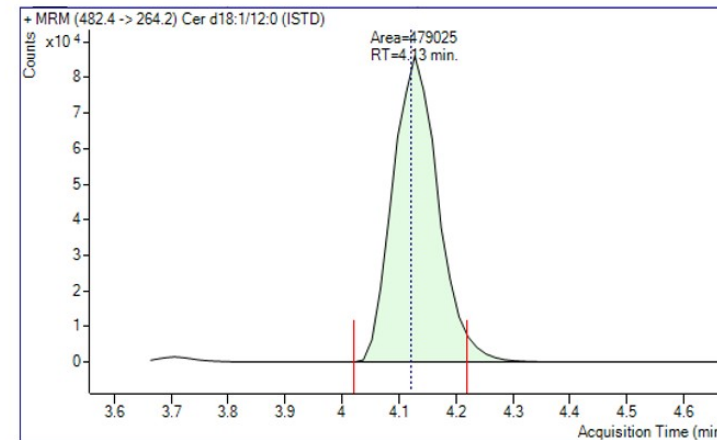


Liquid Chromatography/Mass Spectrometry

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

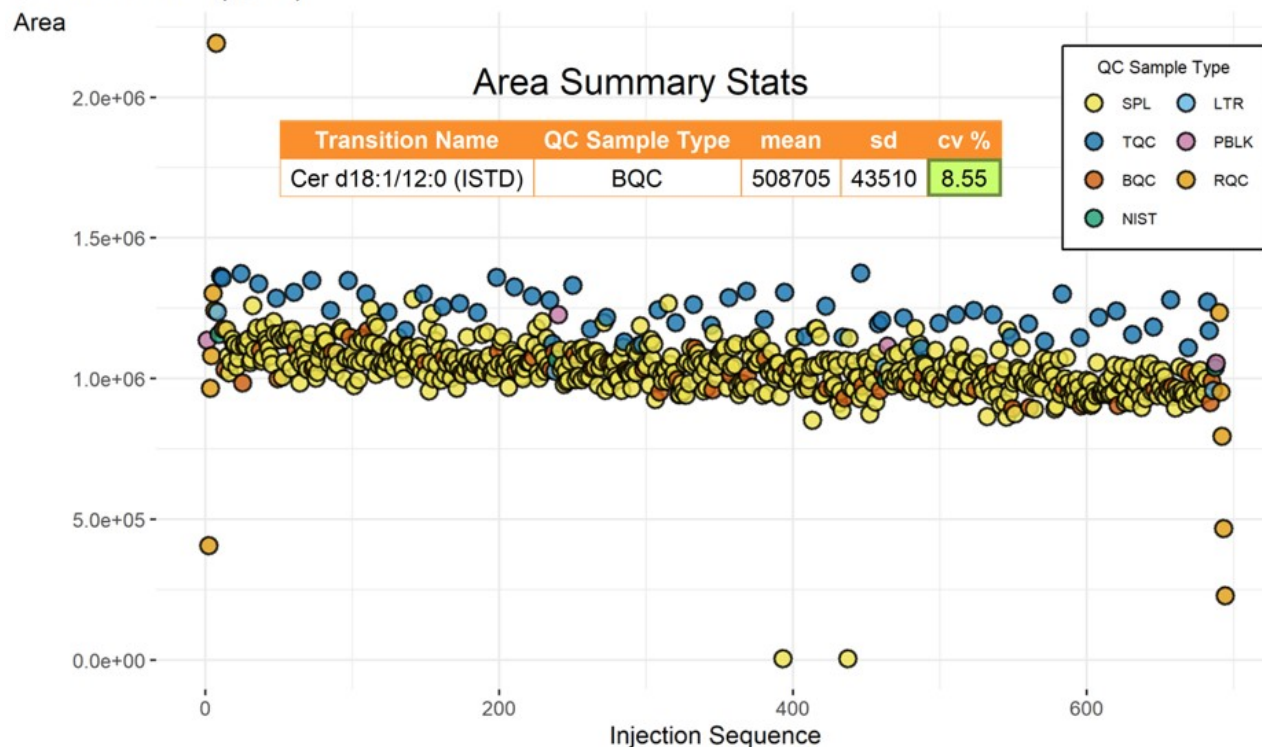
Peak integration



Quality Control (QC) Samples

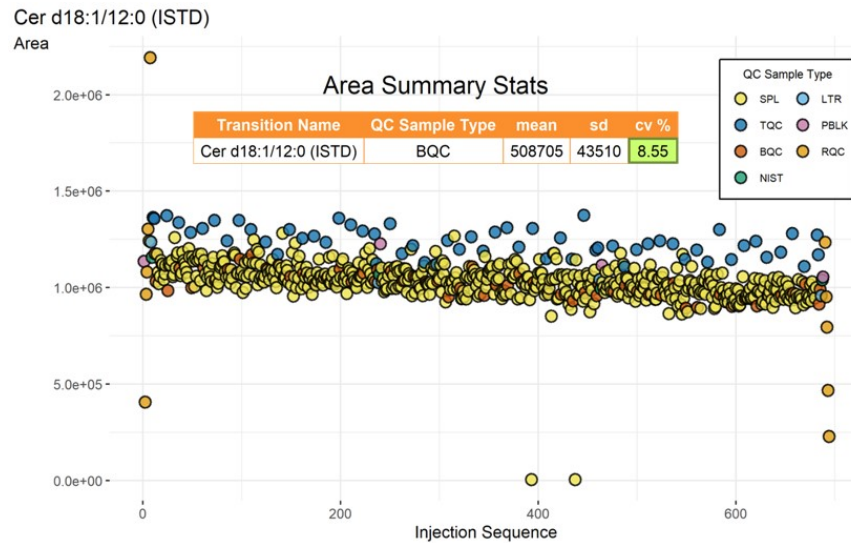
Quality control samples helps to check the severity of variation from external sources such as contaminant ions. We want to keep molecules that gives a low variation on the quality control samples. Plots are exported as pages in a pdf file.

Cer d18:1/12:0 (ISTD)



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

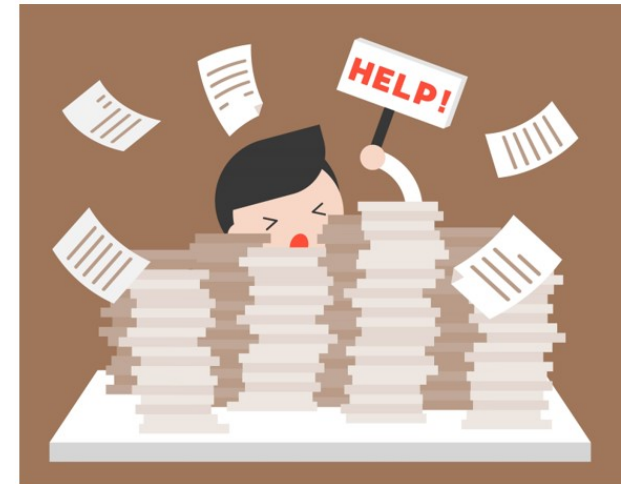
Today's targeted lipidomics workflow can measure up to a few hundred molecules. This gives a pdf file of over 500 pages. Looking at these static plot individually to gain insights is tedious.



> 500



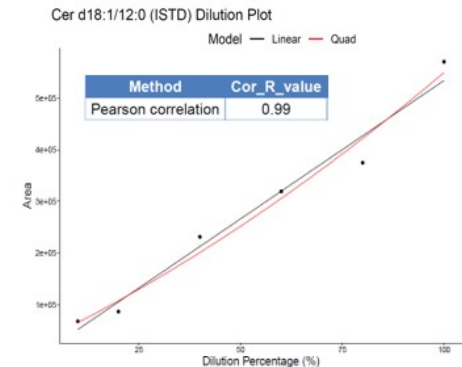
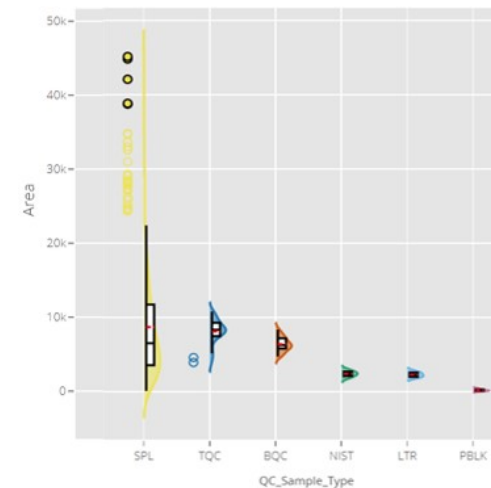
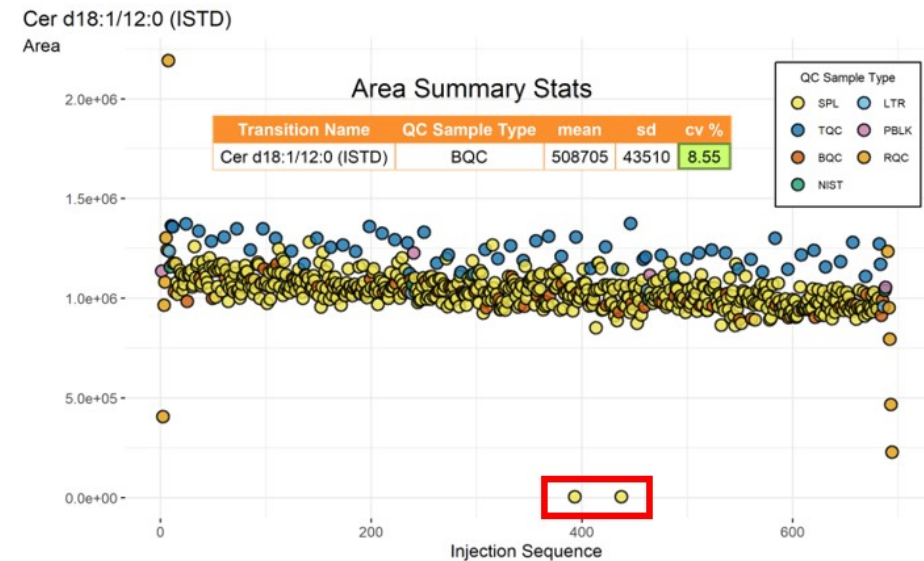
> 500 Pages



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

Out of > 500 molecules,

- Can I have more info about the outlying samples in the red box ?
- How many Ceramides have BQC CV over 20% ?
- Can you plot the same data as a Raincloud Plot ?



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

Using **plotly** to create interactive plots is moving in the right direction.

However, distribution of such results to collaborators/managers remains a challenge.

Interactive plots cannot be stored in pdf files.

Shiny was considered but

- Lack expertises to maintain a secure web server to run **Shiny** applications.
- Cannot expect collaborators/managers to install/run R packages and code to rebuild the **Shiny** application just to view the results.



Data (D:) > Trelliscopejs_In_Quarto_Example > Dilution_Plot_Folder				Search Dilutic
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	



Dilution_Plot_Folder.zip	3/8/2022 5:48 PM	Compressed (zipp...	3,405 KB
--------------------------	------------------	---------------------	----------

Select a Display to Open

no thumbnail

no thumbnail

Dilution_Plot

750 panels, 2022-08-07 15:45

Injection_Sequence_Plot

750 panels, 2022-08-07 15:48

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](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](#)
- [Background](#)
- [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

 Code 

Show All Code

Hide All Code

View Source

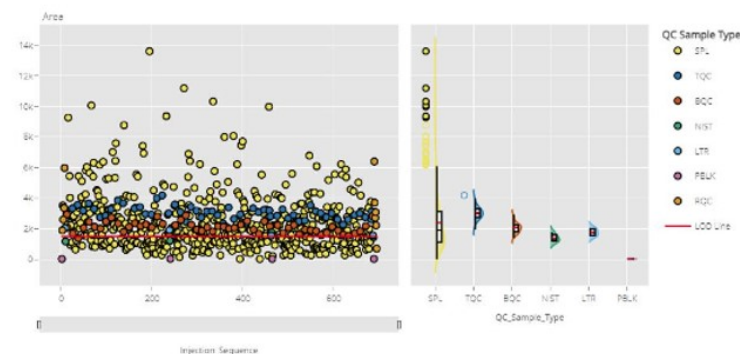
Written by Jeremy Selva    

Introduction

Here is a report showing how to create injection sequence plot and dilution plot using [plotly](#) for each Multiple Reaction Monitoring (MRM) transition in Mass Spectrometry.

Injection Sequence Plot

Dilution Plot



Column 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

File Edit Code View Plots Session Build Debug Profile Tools Help

dilution_plot_table

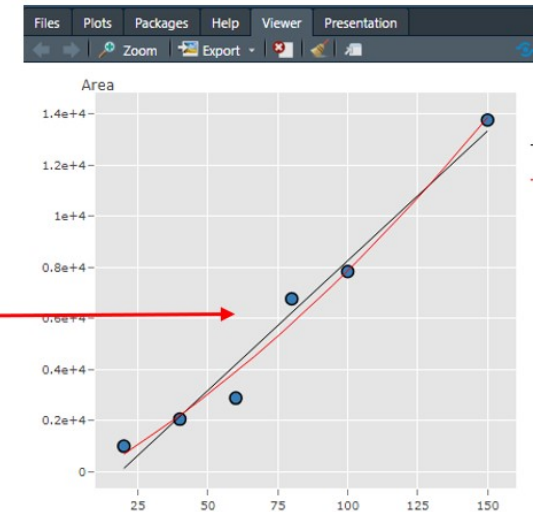
Transition_Name	dilution_panel	Lipid_Name
1 AcylCarnitine 10:0	<code>list(x = list(visdat = list('77644c483021' = funct [...]</code>	CAR 10:0
2 AcylCarnitine 10:1	<code>list(x = list(visdat = list('776423dd5818' = funct [...]</code>	CAR 10:1
3 AcylCarnitine 12:0	<code>list(x = list(visdat = list('776449445435' = 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

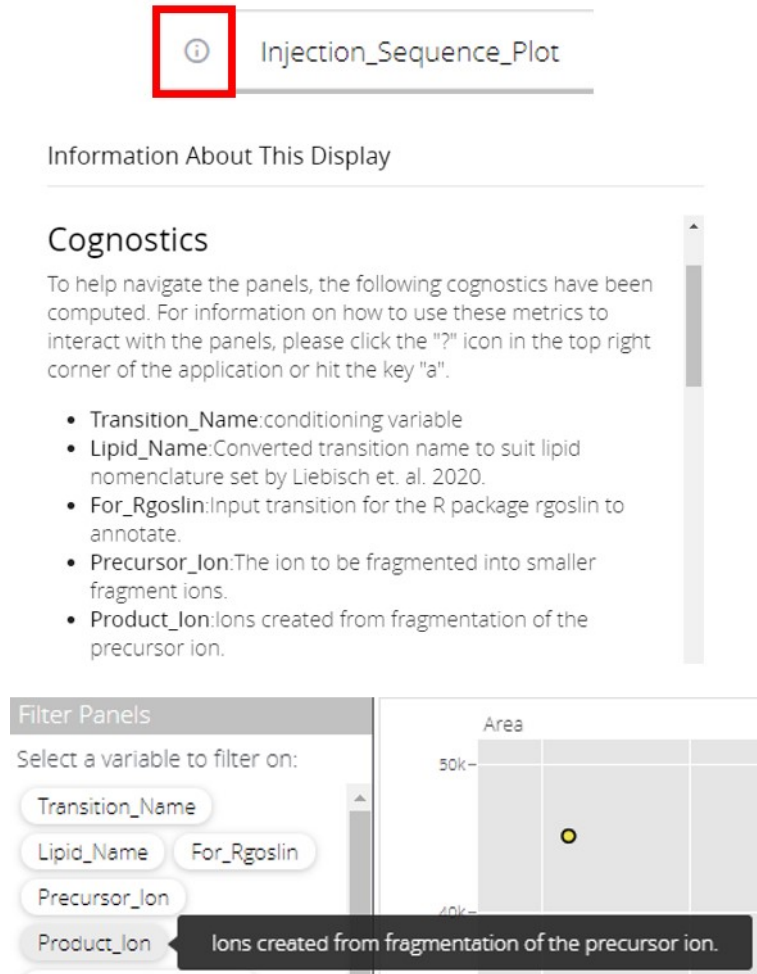
R 4.2.1 · D:/Trelliscopejs_In_Quarto_Example/

```
> dilution_plot_table$dilution_panel[[1]]
```



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.



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.

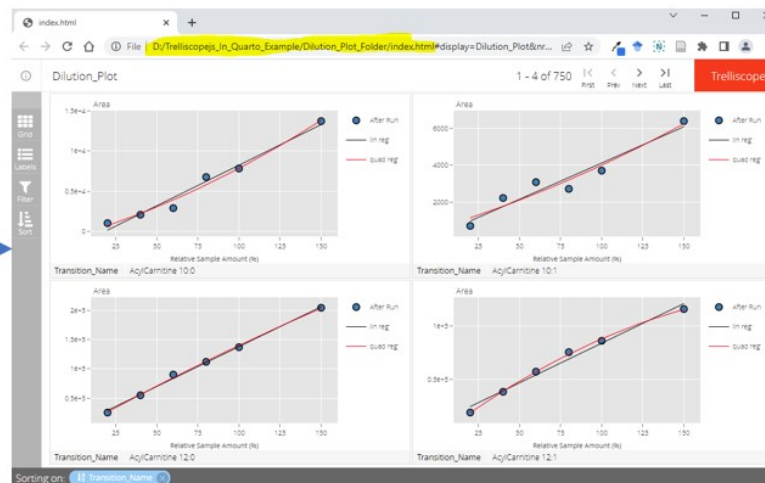
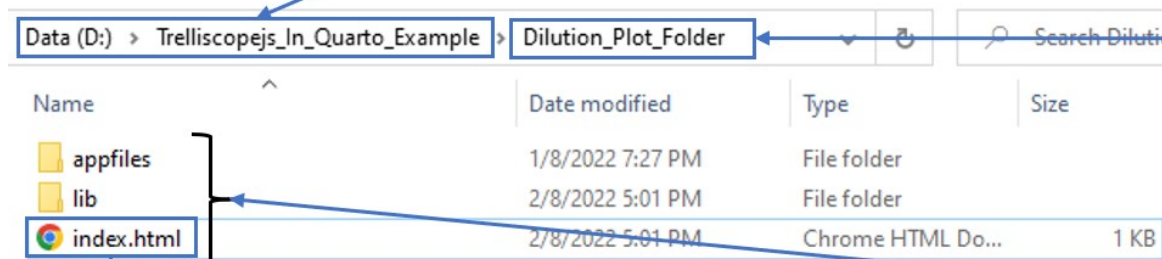
Step 1. Set working directory

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

Step 2. Call script in 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
    )
  )
```



Step 3. Compress Folder

Dilution_Plot_Folder.zip

3/8/2022 5:48 PM

Results Distribution

To export a Quarto document with `trelliscopejs` object.

Step 1. Set working directory

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

Step 2. Render Quarto script



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

```
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
  )
)
```

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
```

Step 3. Compress Highlighted Files/Folders

Quarto_Example.zip

3/8/2022 6:59 PM

Results Distribution

Exported examples are found under the [Releases](#) section in the below [GitHub page](#).'

The screenshot displays the GitHub repository page for `JauntyJJS / Trelliscopejs_In_Quarto_Example`. The repository is public and has 4 stars. The main branch is `main`. The repository contains 1 branch and 3 tags. The file list shows various folders and files, including `Dilution_Plot_Folder`, `Injection_Sequence_Plot_Fol...`, `_extensions`, `data`, `images`, `index_files`, `utils`, `.gitattributes`, and `.gitignore`.

The 'About' section provides information about the repository, including the source code for a Quarto report using Plotly and Trelliscopejs, the repository URL, and the number of stars and forks.

The 'Releases' section shows the latest release, **Version 1.0.2**, which is marked as 'Latest' and was released 9 days ago. A red box highlights this release, and a red arrow points from it to the 'Assets' section.

The 'Assets' section lists 7 assets, including `data.zip`, `Dilution_Plot_Folder.zip`, `docs.zip`, `Injection_Sequence_Plot_Folder.zip`, `Quarto_Example.zip`, `Source code (zip)`, and `Source code (tar.gz)`. A red box highlights the first five assets, which are the exported examples.

Other Resources

trelliscopejs Examples

- <https://ryanhafen.com/blog/pokemon/>
- <https://hafen.github.io/trelliscopejs-demo/mri/>
- <https://hafen.github.io/trelliscopejs-demo/network/>
- <https://www.rostrum.blog/2019/06/20/goat-scope/>

Presentation

- <https://www.rstudio.com/resources/rstudioconf-2017/trelliscopejs/>
- <https://www.youtube.com/watch?v=TgFJrfSEKuQ>
 - See talk at around 54.55
 - [Slides](#)

Hope to see more examples...

Conclusion

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.

