

# Package: chromr (via r-universe)

October 22, 2024

**Type** Package

**Title** Read and Plot FPLC Chromatograms

**Version** 0.2.0

**Author** Brady Johnston

**Maintainer** Brady Johnston <brady.johnston@me.com>

**Description** For reading files from BioRad QuadTec machines and making plots of the chromatograms.

**License** MIT + file LICENSE

**URL** <https://github.com/BradyAJohnston/chromr>

**BugReports** <https://github.com/BradyAJohnston/chromr/issues>

**Imports** dplyr, ggplot2, glue, janitor, lifecycle, magrittr, purrr, readr, rlang, scales, stringr, tibble, tidyr

**Suggests** covr, plotly, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.2.1

**Repository** <https://bradyajohnston.r-universe.dev>

**RemoteUrl** <https://github.com/bradyajohnston/chromr>

**RemoteRef** HEAD

**RemoteSha** 1785118ba34cf8942b5701faae442e1f2d1d50e8

## Contents

check_column_exist . . . . .	2
chrom_add_volume . . . . .	2
chrom_append_run . . . . .	3
chrom_find_data_start_line . . . . .	4
chrom_get_meta_quadtech . . . . .	5
chrom_plot . . . . .	5

chrom_plot_fractions . . . . .	6
chrom_read_ngc . . . . .	7
chrom_read_quadtech . . . . .	8
interpolate_column . . . . .	9
rename_columns . . . . .	9

## Index 10

check\_column\_exist      *Check if Column is Present in DataFrame Ignoring Case*

### Description

Check if Column is Present in DataFrame Ignoring Case

### Usage

```
check_column_exist(data, name)
```

### Arguments

data	Dataframe
name	string of column name

### Value

Logical TRUE/FALSE if the column exists.

chrom\_add\_volume      *Add Volume Column From Time Units*

### Description

Adds a column with calculated volumes from the time column given a particular flow rate. Currently only constant flow rates are supported.

### Usage

```
chrom_add_volume(.data, flow_rate = 0.5, time = "second")
```

### Arguments

.data	Data frame or tibble with a column called 'time'.
flow_rate	Flow rate in ml/min.
time	Time unit when exported.

**Value**

a [tibble][tibble::tibble-package]

**Examples**

```
f1 <- system.file(
  "extdata",
  "sec_no_volume.txt",
  package = "chromr"
)
# read just the data
dat <- f1 %>%
  chrom_read_quadtech()
dat
# add a volume given a constant flow rate
dat %>%
  chrom_add_volume(0.3)
```

---

chrom_append_run	<i>Append a Chromatogram to Another Run</i>
------------------	---

---

**Description**

Sums the volume and the time columns. Optionally continues the numbering of the fractions after the previous runs.

**Usage**

```
chrom_append_run(.data, ..., adjust_fractions = FALSE)
```

**Arguments**

.data	A data frame created through ‘chrom_read_*()’
...	Runs created through ‘chrom_read_*()’
adjust_fractions	Logical. FALSE keeps the fraction numbering as-is. TRUE starts the number of the subsequent fractions by adding the previous fractions.

**Value**

a [tibble][tibble::tibble-package] which combines the given runs.

## Examples

```
f11 <- system.file("extdata",
  "20220809_SFPQf1_TEVdig_S200_part1.TXT",
  package = "chromr"
)
f12 <- system.file("extdata",
  "20220809_SFPQf1_TEVdig_S200_part2.TXT",
  package = "chromr"
)
df1 <- chrom_read_quadtech(f11)
df2 <- chrom_read_quadtech(f12)

df1 %>%
  chrom_append_run(df2)
```

---

chrom\_find\_data\_start\_line

*Find the Line Where Data Begins*

---

## Description

Finds the line where the tabular data begins. This is then used to start the reading of the data with 'readr::read\_csv()' and end the reading of the metadata.

## Usage

```
chrom_find_data_start_line(file, n_lines = 50)
```

## Arguments

file	File path to the file to read.
n_lines	Number of lines to search for the start of the data.

## Value

Single integer of the start of the data.

---

`chrom_get_meta_quadtech`*Reads Metadata from QuadTech Chromatogram*

---

**Description**

Reads Metadata from QuadTech Chromatogram

**Usage**

```
chrom_get_meta_quadtech(file, start_line)
```

**Arguments**

<code>file</code>	Path to the chromatogram file.
<code>start_line</code>	Start of the data and thus end of the metadata. Determined with <code>'chromr::chrom_find_data_start_line()'</code>

**Value**

a [tibble][tibble::tibble-package]

---

`chrom_plot`*Plot a Chromatogram*

---

**Description**

Plot a Chromatogram

**Usage**

```
chrom_plot(data, xlim = NULL, ylim = NULL)
```

**Arguments**

<code>data</code>	Data frame that contains columns for wavelength ( <code>'wl'</code> ), volume ( <code>'volume'</code> ) and absorbance ( <code>'abs'</code> ).
<code>xlim</code>	Limits for the x axis.
<code>ylim</code>	Limits for the y axis.

**Value**

a `'ggplot2::ggplot()'` plot.

**Examples**

```
f1 <- system.file(
  "extdata",
  "sec_no_volume.txt",
  package = "chromr"
)

f1 %>%
  chrom_read_quadtech() %>%
  chrom_add_volume(0.3) %>%
  chrom_plot(xlim = c(0, 3), ylim = c(NA, 0.01))
```

---

chrom\_plot\_fractions *Plot Chromatogram with Fractions*

---

**Description**

Plot Chromatogram with Fractions

**Usage**

```
chrom_plot_fractions(
  data,
  wl_frac = 280,
  fractions = TRUE,
  frac_include = "all",
  frac_labelling = 5,
  frac_text_size = 3,
  frac_text_adjust = 1.3
)
```

**Arguments**

data	Datafram containing values.
wl_frac	Wavelengths to show the fractionation scheme for.
fractions	Logical, whether to include fractions on the plot.
frac_include	Specific fractions to include. Either "all" for all fractions, or a numeric vector of length 2, specifying the limits for the fractions to be included (e.g. c(10, 30) includes fractions from 10 till 30, including both).
frac_labelling	How often to label the fractions. Every \$n_th\$ fraction is labelled.
frac_text_size	Size of the labels for the fractionation.
frac_text_adjust	'vjust' for the labels for the fractionation.

**Value**

a ggplot object.

**Examples**

```
f11 <- system.file("extdata",
  "20220809_SFPQf1_TEVdig_S200_part1.TXT",
  package = "chromr"
)
f12 <- system.file("extdata",
  "20220809_SFPQf1_TEVdig_S200_part2.TXT",
  package = "chromr"
)
df1 <- chrom_read_quadtech(f11)
df2 <- chrom_read_quadtech(f12)
dat <- chrom_append_run(df1, df2)
chrom_plot_fractions(dat, wl_frac = c(280, 488))
```

---

chrom_read_ngc	<i>Read .csv Chromatogram from the BioRad NGC</i>
----------------	---

---

**Description**

Read .csv Chromatogram from the BioRad NGC

**Usage**

```
chrom_read_ngc(file, skip = 1)
```

**Arguments**

file	File path to the '.csv' file.
skip	Number of lines to skip before begin reading. Usually 1 line.

**Value**

a [tibble][tibble::tibble-package]

**Examples**

```
f1 <- system.file(
  "extdata",
  "ngc_sec.csv",
  package = "chromr"
)

dat <- chrom_read_ngc(f1)
dat
```

---

chrom\_read\_quadtech    *Read BioRad QuadTech Chromatogram Files*

---

## Description

Read BioRad QuadTech Chromatogram Files

## Usage

```
chrom_read_quadtech(file, interp_volume = TRUE)
```

## Arguments

`file`                    Exported '.TXT' chromatogram file from the BioRad QuadTech.

`interp_volume`        Logical. If TRUE, interpolates the values in the volume column based on the values in the time column.

## Value

a [tibble][tibble::tibble-package]

## Examples

```
f1 <- system.file("extdata",
  "sec.txt",
  package = "chromr"
)
# just read
f1 %>%
  chrom_read_quadtech()

# read without interpolating volume
f1 %>%
  chrom_read_quadtech(interp_volume = FALSE)
# read then plot
f1 %>%
  chrom_read_quadtech() %>%
  chrom_plot()
```



---

interpolate_column	<i>Interpolate Interpolate Given Column</i>
--------------------	---

---

**Description**

Interpolate Interpolate Given Column

**Usage**

```
interpolate_column(data, col)
```

**Arguments**

data	Dataframe with column to interpolate.
col	Name of the column to interpolate.

**Value**

a [tibble][[tibble::tibble-package]

---

rename_columns	<i>Title</i>
----------------	--------------

---

**Description**

Title

**Usage**

```
rename_columns(data)
```

**Arguments**

data	Dataframe to rename the columns of.
------	-------------------------------------

**Value**

Dataframe with renamed column.

# Index

`check_column_exist`, [2](#)  
`chrom_add_volume`, [2](#)  
`chrom_append_run`, [3](#)  
`chrom_find_data_start_line`, [4](#)  
`chrom_get_meta_quadtech`, [5](#)  
`chrom_plot`, [5](#)  
`chrom_plot_fractions`, [6](#)  
`chrom_read_ngc`, [7](#)  
`chrom_read_quadtech`, [8](#)  
  
`interpolate_column`, [9](#)  
  
`rename_columns`, [9](#)