Package ‘itraxR’

May 8, 2026

Type Package

Title Itrax Data Analysis Tools

Version 1.13.2

Date 2025-12-21

URL https://github.com/tombishop1/itraxR/

BugReports https://github.com/tombishop1/itraxR/issues

Description Parse, trim, join, visualise and analyse data from Itrax sediment core multi-parameter scanners manufactured by Cox Analytical Systems, Sweden. Functions are provided for parsing XRF-peak area files, line-scan optical images, and radiographic images, alongside accompanying metadata. A variety of data wrangling tasks like trimming, joining and reducing XRF-peak area data are simplified. Multivariate methods are implemented with appropriate data transformation.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

LazyData true

Depends dplyr, ggplot2, R (>= 4.1.0), compositions, grid

Imports readr, tiff, janitor, ggcorrplot, rlang, tidyr, broom, tibble, stringr, munsellinterpol, utils, plyr, tidyselect

Suggests magrittr

NeedsCompilation no

Author Thomas Bishop [aut, cre]

Maintainer Thomas Bishop <`tombishopemail@gmail.com`>

Repository CRAN

Date/Publication 2025-12-21 22:50:02 UTC

Contents

CD166_19_S1 2
itrax_correlation 3
itrax_image 4
itrax_import 5
itrax_join 6
itrax_meta 6
itrax_munsell 7
itrax_ordination 8
itrax_qspecsettings 9
itrax_radiograph 9
itrax_reduce 10
itrax_restspectra 11
itrax_section 12
itrax_spectra 13
itrax_sumspectra 14

Index **16**

`CD166_19_S1` *Itrax core scanner data for core CD166_19_S1*

Description

A dataset containing XRF, optical and radiographic images and metadata for ocean core CD166_19_S1. This is a subset of a more complete dataset available.

Usage

```
CD166_19_S1
```

Format

A list containing the following:

xrf a tibble of scan variables

rgb a list containing the optical image matrix and associated metadata

rad a list containing the radiographic image matrix and associated metadata

Source

Bishop, Thomas; Charidemou, Miros (2023): Core Scanning Data from Core CD166/19. PANGAEA, https://doi.pangaea.de/10.1594/PANGAEA.955347

`itrax_correlation` *Calculate a correlation matrix for Itrax result data*

Description

Calculates a correlation matrix for Itrax data results including normalisation and visualisation

Usage

```
itrax_correlation(
  dataframe,
  elementsonly = TRUE,
  zeros = "addone",
  transform = TRUE,
  plot = FALSE
)
```

Arguments

`dataframe`	pass the name of a dataframe parsed using "`itrax_import()`" or "`itrax_join()`"
`elementsonly`	if TRUE, only chemical elements are included. If FALSE, the data is passed unfiltered, otherwise a character vector of desired variable names can be supplied
`zeros`	if "addone", adds one to all values. If "limit", replaces zero values with 0.001. Otherwise a function can be supplied to remove zero values.
`transform`	binary operator that if TRUE will center-log-transform the data, if FALSE will leave the data untransformed. Otherwise, a function can be supplied to transform the data.
`plot`	set to true if a biplot is required as a side-effect

Value

a correlation matrix object

Examples

```
itrax_correlation(CD166_19_S1$xrf, plot = TRUE)
```

`itrax_image` *Read an Itrax Image File*

Description

Reads an Itrax image file and trims it according to the metadata provided.

Usage

```
itrax_image(
  file = "optical.tif",
  meta = "document.txt",
  plot = FALSE,
  trim = TRUE
)
```

Arguments

`file`	defines the name of the datafile to parse
`meta`	defines the relating metadata
`plot`	would you like to create a plot as a side-effect?
`trim`	defines custom trim parameters. The default behaviour uses the limits from the metadata file. Set the false for no trimming, or set the position limits by passing a two element vector.

Value

a matrix of RGB values, and the relevant data from the metadata file relating to the image.

Examples

```
itrax_image(file = system.file("extdata",
                               "CD166_19_S1_optical_lowres.tif",
                               package = "itraxR",
                               mustWork = TRUE),
            meta = system.file("extdata",
                               "CD166_19_S1_xrf_document.txt",
                               package = "itraxR",
                               mustWork = TRUE),
            plot = TRUE)
```

`itrax_import` *Import Itrax core-scanner result file*

Description

Imports and parses data from a results file created by Q-Spec software, part of the Itrax core scanner.

Usage

```
itrax_import(
  filename = "Results.txt",
  depth_top = NA,
  trim_top = 0,
  trim_bottom = 0,
  parameters = "some"
)
```

Arguments

`filename`	defines the name of the datafile to parse
`depth_top`	defines the coring in depth of the top of the core, in mm
`trim_top`	defines the length of any trimming required of data at the top of the core, in mm
`trim_bottom`	defines the length of any trimming required at the bottom of the core, in mm
`parameters`	one of ‘all‘ (leave all parameters), ‘some‘ (remove some less useful parameters)

Value

a tibble of the parsed Itrax data

Examples

```
itrax_import(
  filename = system.file("extdata",
                         "CD166_19_S1_Results.txt",
                         package = "itraxR",
                         mustWork = TRUE),
  depth_top = 0)
```

`itrax_join` *Join two or more Itrax result datasets*

Description

Join two or more Itrax datasets that have been parsed using `"itrax_import()"`

Usage

```
itrax_join(list)
```

Arguments

`list`	a list of dataframes that are parsed Itrax result files — this should have been imported using `"itrax_import()"` and must have a depth variable present. This should take the form `"list(core1 = core1, core2 = core2)"`

Value

a tibble of all the input data

Examples

```
itrax_join(list(core1 = CD166_19_S1$xrf, core2 = CD166_19_S1$xrf))
```

`itrax_meta` *Parse Itrax scan metadata*

Description

Parses the `"document.txt files"` generated from Itrax core scanners

Usage

```
itrax_meta(datafile = "document.txt")
```

Arguments

`datafile`	a `"document.txt files"` generated from an Itrax core scanner

Value

a dataframe of all the parsed input data

Examples

```
itrax_meta(system.file("extdata",
                       "CD166_19_S1_xrf_document.txt",
                       package = "itraxR",
                       mustWork = TRUE))
```

---

itrax_munsell *Convert an Itrax Image File into Munsell Colour*

Description

Reads a colour calibrated Itrax image file and processes it to estimate Munsell colour.

Usage

```
itrax_munsell(image, proportion = 0.1)
```

Arguments

`image`	defines the name of the image file imported using 'itrax_image()'. It is essential that the image has been colour calibrated using a colour card or other method.
`proportion`	defines the width down the centre of the image to use for processing

Value

a table of values

Examples

```
## Not run:
itrax_image(file = system.file("extdata",
                              "CD166_19_S1_optical_lowres.tif",
                              package = "itraxR",
                              mustWork = TRUE),
           meta = system.file("extdata",
                              "CD166_19_S1_xrf_document.txt",
                              package = "itraxR",
                              mustWork = TRUE),
           plot = FALSE) %>%
 magrittr::extract2(1) %>%
 itrax_munsell() %>%
 dplyr::slice_sample(n = 10)

## End(Not run)
```

`itrax_ordination` *Principle Component Analysis on Itrax scan data*

Description

Performs and visualises principle component analysis data from Itrax result data

Usage

```
itrax_ordination(
  dataframe,
  elementsonly = TRUE,
  zeros = "addone",
  transform = TRUE,
  return = "list",
  plot = FALSE
)
```

Arguments

`dataframe`	pass the name of a dataframe parsed using `"itrax_import()"` or `"itrax_join()"`
`elementsonly`	if TRUE, only chemical elements are included. If FALSE, the data is passed unfiltered, otherwise a character vector of desired variable names can be supplied
`zeros`	if "addone", adds one to all values. If "limit", replaces zero values with 0.001. Otherwise a function can be supplied to remove zero values.
`transform`	binary operator that if TRUE will center-log-transform the data, if FALSE will leave the data untransformed. Otherwise, a function can be supplied to transform the data.
`return`	if "pca" the output of `prcomp()` is returned, otherwise "list" is a list including the transformed data, sample scores, and the output of prcomp().
`plot`	set to true if a biplot is required as a side-effect

Value

either an output of `prcomp()`, or a list including the input data

Examples

```
itrax_ordination(CD166_19_S1$xrf, plot = TRUE)
```

`itrax_qspecsettings` *Read a Q-Spec settings file and parse the key-value pairs*

Description

This is used to retrieve settings important elsewhere, for example the mca bin width and offset

Usage

```
itrax_qspecsettings(filename = "Results_settings.dfl")
```

Arguments

`filename`	the *.dfl settings file that relates to the rest of the data

Value

a tibble of the parsed data

Examples

```
itrax_qspecsettings(filename = system.file("extdata",
                                 "Results_settings.dfl",
                                 package = "itraxR",
                                 mustWork = TRUE)
                                 )
```

`itrax_radiograph` *Read an Itrax Radiograph File*

Description

Reads an Itrax radiograph file and trims it according to the metadata provided.

Usage

```
itrax_radiograph(
  file = "radiograph.tif",
  meta = "document.txt",
  plot = FALSE,
  trim = TRUE
)
```

Arguments

`file`	defines the name of the datafile to parse
`meta`	defines the relating metadata
`plot`	would you like to create a plot as a side-effect?
`trim`	defines positions of the trim if required, input is a vector with min and max positions

Value

a matrix of RGB values, and the relevant data from the metadata file relating to the image. Also computes the aspect ratio of the image.

Examples

```
itrax_radiograph(file = system.file("extdata",
                         "CD166_19_S1_radiograph_adj.tif",
                         package = "itraxR",
                         mustWork = TRUE),
                 meta = system.file("extdata",
                         "CD166_19_S1_rad_document.txt",
                         package = "itraxR",
                         mustWork = TRUE),
                 plot = TRUE)
```

`itrax_reduce` *Reduce Itrax XRF data*

Description

Reduces Itrax XRF data into arbitrary chunks using an arbitrary function. This is useful when making direct comparisons between the Itrax XRF data and some other data collected at a lower resolution.

Usage

```
itrax_reduce(
  dataframe,
  names = c(1:length(breaks_lower)),
  breaks_lower,
  breaks_upper,
  fun = mean,
  edges = c(">=", "<"),
  by = NULL
)
```

Arguments

`dataframe`	defines the name of the XRF data to reduce, usually a itraxR::itrax_import like tibble
`names`	optional, a vector of the same length as ‘breaks‘
`breaks_lower`	a vector of the lower limit of each chunk
`breaks_upper`	a vector of the upper limit of each chunk
`fun`	the function to apply in order to reduce the data. Default is mean(), but sd() is also common
`edges`	a vector of length 2 with the upper and lower bound behaviour; can be any of ‘<‘, ‘<=‘, ‘>‘, ‘>=‘
`by`	if contiguous samples of even sizes are required, ‘by‘ defines the chunk size and will automatically generate ‘breaks‘

Value

a tibble with the same number of rows as ‘breaks‘ and the same number of columns as ‘dataframe‘

Examples

```
itrax_reduce(dataframe = CD166_19_S1$xrf, by = 10)
```

`itrax_restspectra` *Make a spectrograph from raw Itrax data spectra files*

Description

Parses a folder full of raw spectra files from an Itrax core scanner and produces a spectral graph of all the data by position

Usage

```
itrax_restspectra(
  foldername = "XRF data",
  parameters = "settings.dfl",
  datapos = 37,
  depthpos = 6,
  plot = TRUE,
  trans = "pseudo_log"
)
```

Arguments

`foldername`	defines the folder where the spectra `"*.spe"` files are located - or the path of the zipped folder where it is stored.
`parameters`	optionally, defines the Q-Spec settings file from which to calculate the channel energies
`datapos`	defines the row at which spectral data begins in the files
`depthpos`	defines the row at which depth data begins is located in the files
`plot`	TRUE/FALSE, selects whether to create a plot as a side-effect
`trans`	transformation applied in the plot - see ‘?ggplot2::scales_colour_gradient()‘ for options

Value

a dataframe of all the spectral data

Examples

```
## Not run: itrax_restspectra("~/itraxBook/CD166_19_(2020)/CD166_19_S1/CD166_19_S1/XRF data")
```

`itrax_section` *Cluster analysis and statistical grouping of Itrax data*

Description

Performs a cluster analysis and automatic statistical grouping of parsed Itrax results data to n groups. Also provides information on the most "representative" (central) of each group. These can be used to develop a sub-sampling regime for calibration using another method.

Usage

```
itrax_section(
  dataframe,
  divisions = 30,
  elementsonly = TRUE,
  zeros = "addone",
  transform = TRUE,
  plot = FALSE
)
```

Arguments

`dataframe`	pass the name of a dataframe parsed using "`itrax_import()`" or "`itrax_join()`" or "`itrax_reduce()`".
`divisions`	the number of groups to slice into - also the number of representative samples returned.
`elementsonly`	if TRUE, only chemical elements are included. If FALSE, the data is passed unfiltered, otherwise a character vector of desired variable names can be supplied.
`zeros`	if "addone", adds one to all values. If "limit", replaces zero values with 0.001. Otherwise a function can be supplied to remove zero values.
`transform`	binary operator that if TRUE will center-log-transform the data, if FALSE will leave the data untransformed. Otherwise, a function can be supplied to transform the data.
`plot`	set to true if a summary plot is required as a side-effect - the input dataset must have a depth or position variable - depth is used preferentially.

Value

the input data with additional columns ‘group‘ and ‘calib_sample‘, and possibly ‘uid‘ if not supplied.

Examples

```
itrax_section(CD166_19_S1$xrf, plot = TRUE)
itrax_section(CD166_19_S1$xrf %>% itrax_reduce(by = 10), plot = TRUE)
```

`itrax_spectra` *Import an individual spectra file*

Description

Sometimes it is helpful to read an individual spectral file for diagnostics

Usage

```
itrax_spectra(filename, parameters = "settings.dfl", plot = TRUE, datapos = 37)
```

Arguments

`filename`	defines the name of the *.spe datafile from the core scanner to parse
`parameters`	optionally defines a relevant Q-Spec settings file in order to compute the energy scale, otherwise channel numbers are reported
`plot`	logical, if TRUE a side-plot is created
`datapos`	defines the row at which spectral data begins in the files

Value

a tibble of the parsed data

Examples

```
itrax_spectra(filename   = system.file("extdata",
                                       "L000676.spe",
                                       package = "itraxR",
                                       mustWork = TRUE),
              parameters = system.file("extdata",
                                         "Results_settings.dfl",
                                         package = "itraxR",
                                         mustWork = TRUE)
              )
```

`itrax_sumspectra` *Create arbitrary sumspectra files*

Description

Parses data from a zipped collection of *.spe files to generate arbitrary sumspectra files from a defined input list

Usage

```
itrax_sumspectra(
  sumspectra_file = "CD166_19_S1/CD166_19_S1/sumspectra.spe",
  input_zip_folder = "CD166_19_S1/CD166_19_S1/XRF Data.zip",
  input_files = c("L000001.spe", "L000002.spe"),
  output_file = "mySumspectra.spe",
  method = mean
)
```

Arguments

`sumspectra_file`	defines the name of the ‘sumspectra.spe‘ file to copy the header from
`input_zip_folder`	defines the zip file to search in for the chosen ‘*.spe‘ files
`input_files`	a character vector of filenames in the format ‘Lnnnnnn.spe‘ to use in the function
`output_file`	defines the filename of the new sumspectra file
`method`	one of ‘mean‘ (useful for generating pseudo ‘*.spe‘ files for calibration) or ‘sum‘ (a more exact recreation of the ‘sumspectra.spe‘ format)

Value

a tab-delimited file in a format compatible with a ‘sumspectra.spe‘ file

Examples

```
## Not run: itrax_sumspectra(sumspectra = "CD166_19_S1/CD166_19_S1/sumspectra.spe",
                          input_zip_folder = "CD166_19_S1/CD166_19_S1/XRF Data.zip",
                          input_files = CD166_19_xrf %>%
                            select(filename, depth) %>%
                            filter(depth >= 100 & depth <= 105) %>%
                            mutate(filename = str_split_i(filename,
                              pattern = "XRF data\\\\",
                              i = 2)
                              ) %>%
                              pull(filename),
                            output_file = "sumspectra_100_105.spe",
                            method = mean)
## End(Not run)
```

Index

∗ **datasets**
    CD166_19_S1, 2

CD166_19_S1, 2

itrax_correlation, 3
itrax_image, 4
itrax_import, 5
itrax_join, 6
itrax_meta, 6
itrax_munsell, 7
itrax_ordination, 8
itrax_qspecsettings, 9
itrax_radiograph, 9
itrax_reduce, 10
itrax_restspectra, 11
itrax_section, 12
itrax_spectra, 13
itrax_sumspectra, 14