Package ‘powdR’

January 10, 2020

Type Package

Title Full Pattern Summation of X-Ray Powder Diffraction Data

Version 1.1.0

Date 2020-01-10

Maintainer Benjamin Butler <benjamin.butler@hutton.ac.uk>


License GPL-2 | file LICENSE

URL http://github.com/benmbutler/powdR

BugReports http://github.com/benmbutler/powdR/issues

Depends R (>= 3.2.0)

Encoding UTF-8

LazyData true

Imports plyr (>= 1.8.4), reshape (>= 0.8.6), plotly (>= 4.7.1), ggplot2 (>= 2.2.1), stats (>= 3.4.3), ggpurbr (>= 0.1.7), shiny (>= 1.0.5), nlns (>= 1.4), shinyWidgets (>= 0.4.3), baseline (>= 1.2), tidyr (>= 0.8), memisc (>= 0.99.17)

Suggests knitr, rmarkdown

RoxygenNote 7.0.2

VignetteBuilder knitr

NeedsCompilation no

Author Benjamin Butler [aut, cre],
     Stephen Hillier [aut],
     Dylan Beaudette [ctb],
     Dennis Eberl [ctb]

Repository CRAN

Date/Publication 2020-01-10 19:30:02 UTC
R topics documented:

afps ................................................................. 2
afps.powdRlib .................................................. 4
bkg ............................................................... 8
fps ............................................................... 9
fps.powdRlib .................................................. 12
minerals ......................................................... 15
minerals_phases ............................................... 16
minerals_xrd ................................................... 17
plot.powdRafps ............................................... 17
plot.powdRbkg ............................................... 18
plot.powdRfps ............................................... 19
plot.powdRlib ............................................... 20
powdR ......................................................... 21
powdRlib ...................................................... 21
rockjock ....................................................... 22
rockjock_mixtures ........................................... 23
run_powdR ...................................................... 24
soils ............................................................. 24
subset.powdRlib ............................................. 25
summarise_mineralogy ....................................... 26

Index 28

afps Automated full pattern summation

Description

afps returns estimates of phase concentrations using automated full pattern summation of X-ray powder diffraction data. It is designed for high-throughput cases involving mineral quantification from large reference libraries. For more details see ?afps.powdRlib.

Usage

afps(lib, ...)

Arguments

lib A powdRlib object representing the reference library. Created using the powdRlib constructor function.

... Other parameters passed to methods e.g. afps.powdRlib

Details

Applies automated full pattern summation to an XRPD measurement to quantify phase concentrations. Requires a powdRlib library of reference patterns with reference intensity ratios in order to derive mineral concentrations.
Value

a list with components:

- **tth**
  - a vector of the 2theta scale of the fitted data

- **fitted**
  - a vector of the fitted XRPD pattern

- **measured**
  - a vector of the original XRPD measurement (aligned)

- **residuals**
  - a vector of the residuals (fitted vs measured)

- **phases**
  - a dataframe of the phases used to produce the fitted pattern

- **phases_grouped**
  - the phases dataframe grouped by phase_name and summed

- **rwp**
  - the Rwp of the fitted vs measured pattern

- **weighted_pure_patterns**
  - a dataframe of reference patterns used to produce the fitted pattern. All patterns have been weighted according to the coefficients used in the fit

- **coefficients**
  - a named vector of coefficients used to produce the fitted pattern

- **inputs**
  - a list of input arguments used in the function call

References


Examples

```r
#Load the minerals library
data(minerals)

# Load the soils data
data(soils)

## Not run:
afps_sand <- afps(lib = minerals,
                  smpl = soils$sandstone,
                  std = "QUA.2",
                  align = 0.2,
                  lod = 0.2,
                  amorphous = "ORG",
                  amorphous_lod = 1)

afps_lime <- afps(lib = minerals,
                  smpl = soils$limestone,
                  std = "QUA.2",
                  align = 0.2,
                  lod = 0.2,
                  amorphous = "ORG",
                  amorphous_lod = 1)
```

align = 0.2,
lod = 0.2,
amorphous = "ORG",
amorphous_lod = 1)

afps_granite <- afps(lib = minerals,
  smpl = soils$granite,
  std = "QUA.2",
  align = 0.2,
  lod = 0.2,
  amorphous = "ORG",
  amorphous_lod = 1)

# Alternatively run all 3 at once using lapply

afps_soils <- lapply(soils, afps,
  lib = minerals,
  std = "QUA.2",
  align = 0.2,
  lod = 0.2,
  amorphous = "ORG",
  amorphous_lod = 1)

# Automated quantification using the rockjock library

data(rockjock)
data(rockjock_mixtures)

# This takes a few minutes to run
rockjock_a1 <- afps(lib = rockjock,
  smpl = rockjock_mixtures$Mix1,
  std = "CORUNDUM",
  align = 0.3,
  lod = 1)

# Quantifying the same sample but defining the internal standard concentration (also takes a few minutes to run):
rockjock_a1s <- afps(lib = rockjock,
  smpl = rockjock_mixtures$Mix1,
  std = "CORUNDUM",
  std_conc = 20,
  align = 0.3,
  lod = 1)

## End(Not run)
afps.powdRlib

Description

afps.powdRlib returns estimates of phase concentrations using automated full pattern summation of X-ray powder diffraction data. It is designed for high-throughput cases involving mineral quantification from large reference libraries.

Usage

## S3 method for class 'powdRlib'
afps(
  lib,
  smpl,
  harmonise,
  solver,
  obj,
  std,
  force,
  std_conc,
  tth_align,
  align,
  manual_align,
  shift,
  shift_res,
  tth_fps,
  lod,
  amorphous,
  amorphous_lod,
  ...
)

Arguments

- **lib**: A powdRlib object representing the reference library. Created using the powdRlib constructor function.
- **smpl**: A data frame. First column is 2theta, second column is counts
- **harmonise**: logical parameter defining whether to harmonise the lib and smpl. Default = TRUE. Harmonises to the intersecting 2theta range at the coarsest resolution available.
- **solver**: The optimisation routine to be used. One of c("BFGS", "Nelder-Mead", "CG" or "L-BFGS-B"). Default = "BFGS".
- **obj**: The objective function to minimise. One of c("Delta", "R", "Rwp"). Default = "Rwp". See Chipera and Bish (2002) and page 247 of Bish and Post (1989) for definitions of these functions.
- **std**: The phase ID (e.g. "QUA.1") to be used as internal standard. Must match an ID provided in the phases parameter.
- **force**: An optional string of phase ID’s or names specifying which phases should be forced to remain throughout the automated full pattern summation. The ID’s or
names supplied must be present within the \texttt{lib$phases$phase_id} or \texttt{lib$phases$phase_name} columns.

\textbf{std\_conc} \hspace{1cm} The concentration of the internal standard (if known) in weight percent. If unknown then use \texttt{std\_conc = NA}, in which case it will be assumed that all phases sum to 100 percent (default).

\textbf{tth\_align} \hspace{1cm} A vector defining the minimum and maximum 2theta values to be used during alignment. If not defined, then the full range is used.

\textbf{align} \hspace{1cm} The maximum shift that is allowed during initial 2theta alignment (degrees). Default = 0.1.

\textbf{manual\_align} \hspace{1cm} A logical operator denoting whether to optimise the alignment within the negative/position 2theta range defined in the \texttt{align} argument, or to use the specified value of the \texttt{align} argument for alignment of the sample to the standards. Default = \texttt{FALSE}, i.e. alignment is optimised.

\textbf{shift} \hspace{1cm} The maximum shift (degrees 2theta) that is allowed during the grid search phases selected from the non-negative least squares. Default = 0).

\textbf{shift\_res} \hspace{1cm} A single integer defining the increase in resolution used during grid search shifting. Higher values facilitate finer shifts at the expense of longer computation. Default = 4.

\textbf{tth\_fps} \hspace{1cm} A vector defining the minimum and maximum 2theta values to be used during automated full pattern summation. If not defined, then the full range is used.

\textbf{lod} \hspace{1cm} Optional parameter used to define the limit of detection (in weight percent) of the internal standard (i.e. the phase provided in the \texttt{std} argument). The \texttt{lod} value is used to estimate the lod of other phases during the fitting process and hence remove reference patterns that are considered below detection limit. Default = 0.1. If \texttt{lod = 0} then limits of detection are not computed.

\textbf{amorphous} \hspace{1cm} A character string of any phase id’s that should be treated as amorphous. Each must match a phase\_id in the phases table of ‘lib’.

\textbf{amorphous\_lod} \hspace{1cm} Optional parameter used to exclude amorphous phases if they are below this specified limit (percent). Must be between 0 and 100. Default = 0.

\textbf{...} \hspace{1cm} other arguments

\textbf{Details}

Applies automated full pattern summation to an XRPD sample to quantify phase concentrations. Requires a \texttt{powdRlib} library of reference patterns with reference intensity ratios in order to derive mineral concentrations.

\textbf{Value}

a list with components:

\textbf{tth} \hspace{1cm} a vector of the 2theta scale of the fitted data

\textbf{fitted} \hspace{1cm} a vector of the fitted XRPD pattern

\textbf{measured} \hspace{1cm} a vector of the original XRPD measurement (aligned)

\textbf{residuals} \hspace{1cm} a vector of the residuals (fitted vs measured)
afps.powdRlib

phases a dataframe of the phases used to produce the fitted pattern
phases_grouped the phases dataframe grouped by phase_name and summed
rwp the Rwp of the fitted vs measured pattern
weighted_pure_patterns a dataframe of reference patterns used to produce the fitted pattern. All patterns have been weighted according to the coefficients used in the fit
coefficients a named vector of coefficients used to produce the fitted pattern
inputs a list of input arguments used in the function call

References


Examples

```r
# Load the minerals library
data(minerals)

# Load the soils data
data(soils)

## Not run:
afps_sand <- afps(lib = minerals,
                    smpl = soils$sandstone,
                    std = "QUA.2",
                    align = 0.2,
                    lod = 0.2,
                    amorphous = "ORG",
                    amorphous_lod = 1)

afps_lime <- afps(lib = minerals,
                   smpl = soils$limestone,
                   std = "QUA.2",
                   align = 0.2,
                   lod = 0.2,
                   amorphous = "ORG",
                   amorphous_lod = 1)

afps_granite <- afps(lib = minerals,
                      smpl = soils$granite,
                      std = "QUA.2",
                      align = 0.2,
                      lod = 0.2,
                      amorphous = "ORG",
                      amorphous_lod = 1)
```
align = 0.2,
lod = 0.2,
amorphous = "ORG",
amorphous_lod = 1)

#Alternatively run all 3 at once using lapply

afps_soils <- lapply(soils, afps,
lib = minerals,
std = "QUA.2",
align = 0.2,
lod = 0.2,
amorphous = "ORG",
amorphous_lod = 1)

#Automated quantification using the rockjock library

data(rockjock)
data(rockjock_mixtures)

#This takes a few minutes to run
rockjock_a1 <- afps(lib = rockjock,
smpl = rockjock_mixtures$Mix1,
std = "CORUNDUM",
align = 0.3,
lod = 1)

#Quantifying the same sample but defining the internal standard
#concentration (also takes a few minutes to run):
rockjock_a1s <- afps(lib = rockjock,
smpl = rockjock_mixtures$Mix1,
std = "CORUNDUM",
std_conc = 20,
align = 0.3,
lod = 1)

## End(Not run)

---

**bkg**

*Fit a background to XRPD data*

**Description**

*bkg* fits a background to X-Ray Powder Diffraction data

**Usage**

*bkg(xrd, lambda, hwi, it, int)*
Arguments

xrd an xy data frame of the data to fit a background to. First column is the 2theta scale, second column is count intensities

lambda second derivative penalty for primary smoothing. Default = 0.5.

hwi Half width of local windows. Default = 25.

it Number of iterations in suppression loop. Default = 50.

int Number of buckets to divide the data into. Default = round(nrow(xrd)/4).

Details

A wrapper for the baseline.fillPeaks in the baseline package.

Value

a list of 3 vectors

tth The 2theta axis of the measurement
counts The count intensities of the measurement
background The fitted background

Examples

data(soils)
fit_bkg <- bkg(soils$granite)

fps

Full pattern summation

Description

fps returns estimates of phase concentrations using full pattern summation of X-ray powder diffraction data. For more details see ?fps.powdRlib.

Usage

fps(lib, ...)

Arguments

lib A powdRlib object representing the reference library. Created using the powdRlib constructor function.

... Other parameters passed to methods e.g. fps.powdRlib
Details

Applies full pattern summation (Chipera & Bish, 2002, 2013; Eberl, 2003) to an XRPD measurement to quantify phase concentrations. Requires a powdRlib library of reference patterns with reference intensity ratios in order to derive mineral concentrations.

Value

a list with components:

- **tth**: a vector of the 2theta scale of the fitted data
- **fitted**: a vector of the fitted XRPD pattern
- **measured**: a vector of the original XRPD measurement (aligned)
- **residuals**: a vector of the residuals (fitted vs measured)
- **phases**: a dataframe of the phases used to produce the fitted pattern
- **phases_grouped**: the phases dataframe grouped by phase_name and summed
- **rwp**: the Rwp of the fitted vs measured pattern
- **weighted_pure_patterns**: a dataframe of reference patterns used to produce the fitted pattern. All patterns have been weighted according to the coefficients used in the fit
- **coefficients**: a named vector of coefficients used to produce the fitted pattern
- **inputs**: a list of input arguments used in the function call

References


Examples

```r
#Load the minerals library
data(minerals)

data(soils)

#Since the reference library is relatively small,
#the whole library can be used at once to get an
#estimate of the phases within each sample.
## Not run:
fps_sand <- fps(lib = minerals,
                smpl = soils$sandstone,
```

fps

refs = minerals$phases$phase_id,
std = "QUA.1",
align = 0.2)

fps_lime <- fps(lib = minerals,
smpl = soils$limestone,
refs = minerals$phases$phase_id,
std = "QUA.1",
align = 0.2)

fps_granite <- fps(lib = minerals,
smpl = soils$granite,
refs = minerals$phases$phase_id,
std = "QUA.1",
align = 0.2)

#Alternatively run all 3 at once using lapply

fps_soils <- lapply(soils, fps,
lib = minerals,
std = "QUA.2",
refs = minerals$phases$phase_id,
align = 0.2)

#Using the rockjock library:

data(rockjock)
data(rockjock_mixtures)

rockjock_1 <- fps(lib = rockjock,
smpl = rockjock_mixtures$Mix1,
refs = c("ORDERED_MICROCLINE",
"LABRADORITE",
"KAOLINITE_DRY_BRANCH",
"MONTMORILLONITE_WYO",
"ILLITE_1M_RM30",
"CORUNDUM"),
std = "CORUNDUM",
align = 0.3)

#Alternatively you can specify the internal standard
#concentration if known:
rockjock_1s <- fps(lib = rockjock,
smpl = rockjock_mixtures$Mix1,
refs = c("ORDERED_MICROCLINE",
"LABRADORITE",
"KAOLINITE_DRY_BRANCH",
"MONTMORILLONITE_WYO",
"ILLITE_1M_RM30",
"CORUNDUM"),
std = "CORUNDUM",
std_conc = 20,
align = 0.3)
fps.powdRlib

---

## End(Not run)

## fps.powdRlib

### Full pattern summation

**Description**

fps.powdRlib returns estimates of phase concentrations using full pattern summation of X-ray powder diffraction data.

**Usage**

```r
## S3 method for class 'powdRlib'
fps(
  lib,
  smpl,
  harmonise,
  solver,
  obj,
  refs,
  std,
  std_conc,
  tth_align,
  align,
  manual_align,
  tth_fps,
  shift,
  shift_res,
  remove_trace,
  ...
)
```

**Arguments**

- `lib`: A powdRlib object representing the reference library. Created using the powdRlib constructor function.
- `smpl`: A data frame. First column is 2theta, second column is counts.
- `harmonise`: logical parameter defining whether to harmonise the `lib` and `smpl`. Default = `TRUE`. Harmonises to the intersecting 2theta range at the coarsest resolution available.
- `solver`: The optimisation routine to be used. One of c("BFGS", "Nelder-Mead", "CG", "L-BFGS-B", or "NNLS"). Default = "BFGS".
The objective function to minimise when "BFGS", "Nelder-Mead", "CG" or "L-BFGS-B" are used as the 'solver' argument. One of c("Delta", "R", "Rwp"). Default = "Rwp". See Chipera and Bish (2002) and page 247 of Bish and Post (1989) for definitions of these functions.

refs
A character string of reference pattern ID’s or names from the specified library. The ID’s or names supplied must be present within the lib$phases$phase_id or lib$phases$phase_name columns.

std
The phase ID (e.g. "QUA.1") to be used as internal standard. Must match an ID provided in the phases parameter.

std_conc
The concentration of the internal standard (if known) in weight percent. If unknown then use std_conc = NA, in which case it will be assumed that all phases sum to 100 percent (default).

tth_align
A vector defining the minimum and maximum 2theta values to be used during alignment. If not defined, then the full range is used.

align
The maximum shift that is allowed during initial 2theta alignment (degrees). Default = 0.1.

manual_align
A logical operator denoting whether to optimise the alignment within the negative/position 2theta range defined in the align argument, or to use the specified value of the align argument for alignment of the sample to the standards. Default = FALSE, i.e. alignment is optimised.

tth_fps
A vector defining the minimum and maximum 2theta values to be used during full pattern summation. If not defined, then the full range is used.

shift
The maximum shift (degrees 2theta) that is allowed during the grid search phases selected from the non-negative least squares. Default = 0).

shift_res
A single integer defining the increase in resolution used during grid search shifting. Higher values facilitate finer shifts at the expense of longer computation. Default = 4.

remove_trace
A single numeric value representing the limit for the concentration of trace phases to be retained, i.e. any mineral with an estimated concentration below 'remove_trace' will be omitted. Default = 0.

Details
Applies full pattern summation (Chipera & Bish, 2002, 2013; Eberl, 2003) to an XRPD sample to quantify phase concentrations. Requires a powdRlib library of reference patterns with reference intensity ratios in order to derive mineral concentrations.

Value
a list with components:

- tth a vector of the 2theta scale of the fitted data
- fitted a vector of the fitted XRPD pattern
- measured a vector of the original XRPD measurement (aligned)
residuals  a vector of the residuals (fitted vs measured)
phases    a dataframe of the phases used to produce the fitted pattern
phases_grouped the phases dataframe grouped by phase_name and summed
rwp       the Rwp of the fitted vs measured pattern
weighted_pure_patterns a dataframe of reference patterns used to produce the fitted pattern. All patterns have been weighted according to the coefficients used in the fit
coefficients a named vector of coefficients used to produce the fitted pattern
inputs    a list of input arguments used in the function call

References


Examples

#Load the minerals library
data(minerals)

# Load the soils data
data(soils)

#Since the reference library is relatively small, #the whole library can be used at once to get an #estimate of the phases within each sample. ## Not run:
fps_sand <- fps(lib = minerals,
               smpl = soils$sandstone,
               refs = minerals$phases$phase_id,
               std = "QUA.1",
               align = 0.2)

fps_lime <- fps(lib = minerals,
                smpl = soils$limestone,
                refs = minerals$phases$phase_id,
                std = "QUA.1",
                align = 0.2)

fps_granite <- fps(lib = minerals,
                    smpl = soils$granite,
refs = minerals$phases$phase_id,
std = "QUA.1",
align = 0.2)

#Alternatively run all 3 at once using lapply

fps_soils <- lapply(soils, fps,
lib = minerals,
std = "QUA.2",
refs = minerals$phases$phase_id,
align = 0.2)

#Using the rockjock library:

data(rockjock)
data(rockjock_mixtures)

rockjock_1 <- fps(lib = rockjock,
smpl = rockjock_mixtures$Mix1,
refs = c("ORDERED_MICROCLINE",
"LABRADORITE",
"KAOLINITE_DRY_BRANCH",
"MONTMORILLONITE_WYO",
"ILLITE_1M_RM30",
"CORUNDUM"),
std = "CORUNDUM",
align = 0.3)

#Alternatively you can specify the internal standard
#concentration if known:
rockjock_1s <- fps(lib = rockjock,
smpl = rockjock_mixtures$Mix1,
refs = c("ORDERED_MICROCLINE",
"LABRADORITE",
"KAOLINITE_DRY_BRANCH",
"MONTMORILLONITE_WYO",
"ILLITE_1M_RM30",
"CORUNDUM"),
std = "CORUNDUM",
std_conc = 20,
align = 0.3)

## End(Not run)
**Description**

This `powdRlib` object, built using the `powdRlib` constructor function, contains a range of measured XRPD data (Cu K-alpha radiation) along with their reference intensity ratios. The library can be used with the `soils` example data for full pattern summation.

**Usage**

```r
minerals
```

**Format**

A list of 3

- `xrd` A dataframe of all xrd data (counts only). Column names denote the reference sample
- `tth` A vector of 2theta intervals of all measurements in the library
- `phases` A dataframe the phase ID’s, names and reference intensity ratios (RIR)

**minerals_phases**

*A table of associated data for the minerals_xrd table, which can be be combined with a xrd data table to create a powdRlib object when using the powdRlib constructor function. Use the same layout to create custom reference libraries.*

**Description**

A table of associated data for the minerals_xrd table, which can be be combined with a xrd data table to create a powdRlib object when using the powdRlib constructor function. Use the same layout to create custom reference libraries.

**Usage**

```r
minerals_phases
```

**Format**

A 3 column dataframe

- The first column is a character string defining the unique mineral ID’s that should match those defined as column names of the minerals table (e.g. `minerals_xrd`).
- The second column is a character string defining the mineral group that each reference pattern belongs to.
- The third column is a numeric vector defining the reference intensity ratios of each reference pattern.
A table of 14 reference patterns and their corresponding two theta scale that can be combined with the minerals_phases table to create a powdRlib object when using the powdRlib constructor function. Use the same layout to create custom reference libraries.

Description

A table of 14 reference patterns and their corresponding two theta scale that can be combined with the minerals_phases table to create a powdRlib object when using the powdRlib constructor function. Use the same layout to create custom reference libraries.

Usage

minerals_xrd

Format

A dataframe

The first column defines the two theta scale, and remaining columns are individual reference patterns of pure minerals or amorphous phases. Each column title should be a unique mineral ID.

plot.powdRafps

Plotting elements of a powdRafps object

Description

plot.powdRafps is designed to provide easy, adaptable plots of full pattern summation outputs produced from afps.

Usage

## S3 method for class 'powdRafps'
plot(x, wavelength, interactive, ...)

Arguments

x a powdRafps object

wavelength One of "Cu", "Co" or a custom numeric value defining the wavelength (in Angstroms). Used to compute d-spacings. When "Cu" or "Co" are supplied, wavelengths of 1.54056 or 1.78897 are used, respectively.

interactive logical. If TRUE then the output will be an interactive ggplotly object. If FALSE then the output will be a ggplot object.

... other arguments
Details

The only mandatory argument is `x`, which must be a `powdRlib` object. When seeking to inspect the results from full pattern summation, interactive plots are particularly useful and can be specified with the `interactive` argument.

Examples

```r
# Load the minerals library
data(minerals)

# Load the soils data
data(soils)

## Not run:
afps_sand <- afps(lib = minerals,
                   smpl = soils$sandstone,
                   std = "QUA.1",
                   amorphous = "ORG",
                   align = 0.2,
                   lod = 0.1)

plot(afps_sand, wavelength = "Cu")
plot(afps_sand, wavelength = "Cu", interactive = TRUE)

## End(Not run)
```

---

**plot.powdRbkg**

*Plotting a `powdRbkg` object*

Description

`plot.powdRbkg` is designed to provide quick plots to inspect the fitted backgrounds obtained from `bkg`.

Usage

```r
## S3 method for class 'powdRbkg'
plot(x, interactive, ...)
```

Arguments

- `x`: a `powdRlib` object
- `interactive`: Logical. If `TRUE` then the output will be an interactive `ggplotly` object. If `FALSE` then the output will be a `ggplot` object.
- `...`: other arguments
plot.powdRfps

Details

The only mandatory argument is x, which must be a powdRbkg object. Plots can be made interactive using the logical interactive argument.

Examples

# Load the minerals library
data(minerals)

## Not run:
plot(minerals, interactive = TRUE)

## End(Not run)

Description

plot.powdRfps is designed to provide easy, adaptable plots of full pattern summation outputs produced from fps.

Usage

## S3 method for class 'powdRfps'
plot(x, wavelength, interactive, ...)

Arguments

- x: a powdRfps object
- wavelength: One of "Cu", "Co" or a custom numeric value defining the wavelength (in Angstroms). Used to compute d-spacings. When "Cu" or "Co" are supplied, wavelengths of 1.54056 or 1.78897 are used, respectively.
- interactive: logical. If TRUE then the output will be an interactive ggplotly object. If FALSE then the output will be a ggplot object.
- ...: other arguments

Details

The only mandatory argument is x, which must be a powdRlib object. When seeking to inspect the results from full pattern summation, interactive plots are particularly useful and can be specified with the interactive argument.
Examples

```r
#Load the minerals library
data(minerals)

# Load the soils data
data(soils)

## Not run:
fps_sand <- fps(lib = minerals, 
               smpl = soils$sandstone, 
               refs = minerals$phases$phase_id, 
               std = "QUA.1", 
               align = 0.2)

plot(fps_sand, wavelength = "Cu")
plot(fps_sand, wavelength = "Cu", interactive = TRUE)

## End(Not run)
```

---

**plot.powdRlib** *Plotting elements of a powdRlib object*

**Description**

plot.powdRlib is designed to provide easy, adaptable plots of an XRPD reference library built using the powdRlib constructor function.

**Usage**

```r
## S3 method for class 'powdRlib'
plot(x, wavelength, refs, interactive, ...)
```

**Arguments**

- `x` a powdRlib object
- `wavelength` One of "Cu", "Co" or a custom numeric value defining the wavelength (in Angstroms). Used to compute d-spacings. When "Cu" or "Co" are supplied, wavelengths of 1.54056 or 1.78897 are used, respectively.
- `refs` a character string of reference pattern id’s to be plotted
- `interactive` Logical. If TRUE then the output will be an interactive ggplotly object. If FALSE then the output will be a ggplot object.
- `...` other arguments

**Details**

The only mandatory argument is `x`, which must be a powdRlib object. Plots can be made interactive using the logical `interactive` argument.
Examples

```r
# Load the minerals library
data(minerals)
## Not run:
plot(minerals, wavelength = "Cu", refs = "ALB")
plot(minerals, wavelength = "Cu", refs = "ALB", interactive = TRUE)
## End(Not run)
```

Description

An implementation of the full pattern summation approach to quantitative mineralogy from X-ray powder diffraction data (Chipera & Bish, 2002, 2013; Eberl, 2003).

Author(s)

Benjamin Butler, The James Hutton Institute, Aberdeen, UK

References


powdRlib

Create an XRD reference library

Description

A constructor function for creating a `powdRlib` object from two tables of data. The resulting reference library object is required when using `fps`.

Usage

```
powdRlib(xrd_table, phases_table)
```
**Arguments**

- **xrd_table**: A data frame of the count intensities of the xrd reference patterns, all scaled to same maximum intensity, with their 2theta axis as the first column. Must all be on the same 2theta scale.
- **phases_table**: A data frame of the required data (phase ID, phase name, and reference intensity ratio) for each reference pattern.

**Value**

A list with components:

- **xrd**: a data frame of reference patterns
- **tth**: a vector of the 2theta measurement intervals
- **phases**: a data frame of the id, name and reference intensity ratio of each reference pattern

**Examples**

```r
#load an example xrd_table
data(minerals_xrd)
#load an example phases_table
data(minerals_phases)

#Create a reference library object
xrd_lib <- powdRlib(xrd_table = minerals_xrd,
                      phases_table = minerals_phases)
```

---

**rockjock**

**RockJock reference library**

**Description**

A powdRlib object of 168 pure reference patterns from the RockJock library (Cu K-alpha radiation) along with reference intensity ratios. Can be used with the fps() and afps() functions for quantitative analysis. Example mixtures for testing the rockjock library with known concentrations are available in the rockjock_mixtures data. See ?rockjock_mixtures.

**Usage**

- rockjock

**Format**

A list of 3 components

- **xrd**: A dataframe of all xrd data (counts only). Column names denote the reference sample
- **tth**: A vector of 2theta intervals of all measurements in the library
- **phases**: A dataframe the phase ID’s, names and reference intensity ratios (RIR)
**rockjock_mixtures**

**Author(s)**

Dennis Eberl

**References**

[https://pubs.er.usgs.gov/publication/ofr200378](https://pubs.er.usgs.gov/publication/ofr200378)

---

**rockjock_mixtures**  
*RockJock synthetic mixtures*

**Description**

A list containing 8 XRPD measurements (Cu K-alpha radiation) of synthetic mixtures that can be used to assess accuracy of quantitative analysis from the `fps()` and `afps()` functions. The mixtures contain various amounts of quartz (QUARTZ standard in of the rockjock library), K-feldspar (ORDERED_MICROCLINE), plagioclase (LABRADORITE), kaolinite (KAOLINITE_DRY_BRANCH), dioctahedral smectite (MONTMORILLIONITE_WYO), illite (ILLITE_1M_RM30) and corundum (CORUNDUM).

**Usage**

`rockjock_mixtures`

**Format**

A list of 8 components, each comprised of two columns. Column `tth` specifies the 2theta axis and `counts` specifies the count intensities.

**Mix1** Contains: 4% K-feldspar, 20% plagioclase, 12% kaolinite, 36% dioctahedral smectite, 8% illite and 20% corundum.

**Mix2** Contains: 4% quartz, 8% K-feldspar, 36% plagioclase, 20% kaolinite, 12% illite and 20% corundum.

**Mix3** Contains: 8% quartz, 12% K-feldspar, 36% kaolinite, 4% dioctahedral smectite, 20% illite and 20% corundum.

**Mix4** Contains: 12% quartz, 20% K-feldspar, 4% plagioclase, 8% dioctahedral smectite, 36% illite and 20% corundum.

**Mix5** Contains: 20% quartz, 36% K-feldspar, 8% plagioclase, 4% kaolinite, 12% dioctahedral smectite and 20% corundum.

**Mix6** Contains: 36% quartz, 12% plagioclase, 8% kaolinite, 20% dioctahedral smectite, 4% illite and 20% corundum.

**Mix7** Contains: 8% K-feldspar, 40% plagioclase, 4% kaolinite, 12% dioctahedral smectite, 16% illite and 20% corundum.

**Mix8** Contains: 8% quartz, 4% K-feldspar, 4% plagioclase, 24% dioctahedral smectite, 40% illite and 20% corundum.
Author(s)

Dennis Eberl

References

https://pubs.er.usgs.gov/publication/ofr200378

---

run_powdR

*Run the powdR shiny app*

---

Description

A wrapper for `runApp` to start the Shiny app for powdR.

Usage

```r
run_powdR(...)
```

Arguments

... further arguments to pass to `runApp`

Examples

```r
## Not run:
run_powdR()

## End(Not run)
```

---

soils

*Example soil XRPD data*

---

Description

3 soil samples from different parent materials measured by XRPD (Cu K-alpha radiation)

Usage

```r
soils
```
subset.powdRlib

Format
A list of 3 dataframes (named according to rock type), with each dataframe containing two columns of:

- **tth** The 2theta measurement intervals
- **counts** The count intensities

Description
subset.powdRlib is designed to provide an easy way of subsetting a powdRlib object by defining the phase ID’s that the user wishes to either keep or remove.

Usage
```r
## S3 method for class 'powdRlib'
subset(x, refs, mode, ...)
```

Arguments
- **x** a powdRlib object.
- **refs** a string of the phase ID’s or names of reference patterns to be subset. The ID’s or names supplied must be present within the lib$phases$phase_id or lib$phases$phase_name columns.
- **mode** denotes whether the phase ID’s or names defined in the refs argument are retained ("keep") or removed ("remove").
- **...** other arguments

Value
a powdRlib object.

Examples
```r
# Load the minerals library
data(minerals)

minerals_keep <- subset(minerals,
                       refs = c("QUA.1", "QUA.2"),
                       mode = "keep")

minerals_remove <- subset(minerals,
                        refs = c("QUA.1", "QUA.2"),
                        mode = "remove")
```
summarise_mineralogy  

Summarise the mineralogy from multiple powdRfps and powdRafps outputs

Description

summarise_mineralogy creates a summary table of quantified mineral concentrations across a given dataset using list of multiple powdRfps or powdRafps derived from fps() and afps(), respectively.

Usage

summarise_mineralogy(x, type, order, rwp)

Arguments

- **x**: a list of powdRfps or powdRafps objects.
- **type**: a string specifying whether the table uses all phase ID’s, or summarises them according to the phase name. One of "all" or "grouped".
- **order**: a logical operator denoting whether the columns of the resulting summary table are ordered in descending order according to the summed abundance of each phase across the dataset.
- **rwp**: a logical operator denoting whether to include the Rwp as the final column in the output. This provides an objective measure of the difference between the fitted and measured patterns.

Value

a dataframe

Examples

data(minerals)
data(soils)

```r
# Not run:
multiple_afps <- lapply(soils, afps,
lib = minerals,
std = "QUA.1",
align = 0.2,
lod = 0.1,
amorphous = "ORG",
amorphous_lod = 1)

sm1 <- summarise_mineralogy(multiple_afps,
type = "all",
order = TRUE)
```


sm2 <- summarise_mineralogy(multiple_afps,
                        type = "grouped",
                        order = TRUE)

sm3 <- summarise_mineralogy(multiple_afps,
                        type = "grouped",
                        order = TRUE,
                        rwp = TRUE)

## End(Not run)
Index

*Topic datasets
    minerals, 15
    minerals_phases, 16
    minerals_xrd, 17
    rockjock, 22
    rockjock_mixtures, 23
    soils, 24

    afps, 2, 17
    afps.powdRlib, 4

    bkg, 8

    fps, 9, 19, 21
    fps.powdRlib, 12

    minerals, 15
    minerals_phases, 16
    minerals_xrd, 17

    plot.powdRafps, 17
    plot.powdRbkg, 18
    plot.powdRfps, 19
    plot.powdRlib, 20
    powdR, 21
    powdRlib, 21

    rockjock, 22
    rockjock_mixtures, 23
    run_powdR, 24
    runApp, 24

    soils, 24
    subset.powdRlib, 25
    summarise_mineralogy, 26