Package 'powdR'

January 15, 2021

Type Package

Title Full Pattern Summation of X-Ray Powder Diffraction Data

Version 1.2.5

Date 2021-01-15

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

Description Full pattern summation of X-ray powder diffraction data as described in Chipera and Bish (2002) <doi:10.1107/S0021889802017405>. Derives quantitative estimates of crystalline and amorphous phase concentrations in complex mixtures.

License GPL-2 | file LICENSE

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

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

Depends R (>= 3.2.0)

Encoding UTF-8

LazyData true

Imports plyr (>= 1.8.6), reshape (>= 0.8.8), plotly (>= 4.9.2.1), ggplot2 (>= 3.3.0), stats (>= 3.4.3), ggpubr (>= 0.2.5), shiny (>= 1.4.0.2), DT (>= 0.13), nnls (>= 1.4), shinyWidgets (>= 0.5.1), baseline (>= 1.2), tidyr (>= 1.0.2)

Suggests knitr, rmarkdown

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

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

Repository CRAN

Date/Publication 2021-01-15 18:00:02 UTC

R topics documented:

`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
`obj`	named vector of the objective parameters summarising the quality of the fit
`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

Chipera, S.J., Bish, D.L., 2013. Fitting Full X-Ray Diffraction Patterns for Quantitative Analysis: A Method for Readily Quantifying Crystalline and Disordered Phases. Adv. Mater. Phys. Chem. 03, 47-53. doi:10.4236/ampc.2013.31A007

Chipera, S.J., Bish, D.L., 2002. FULLPAT: A full-pattern quantitative analysis program for X-ray powder diffraction using measured and calculated patterns. J. Appl. Crystallogr. 35, 744-749. doi:10.1107/S0021889802017405

Eberl, D.D., 2003. User's guide to RockJock - A program for determining quantitative mineralogy from powder X-ray diffraction data. Boulder, CA.

Examples

```
#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)

#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 *Automated full pattern summation*

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,
  refs,
  std,
  force,
  std_conc,
  normalise,
  tth_align,
  align,
  manual_align,
  shift,
  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 using natural splines.
solver	The optimisation routine to be used. One of c("BFGS","Nelder-Mead",or "CG"). 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.
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. If missing from the function call then all phases in the reference library will be used.
std	The phase ID (e.g. "QUA.1") to be used as internal standard. Must match an ID provided in the refs 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 lib$phases$phase_id or lib$phases$phase_name columns.
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).
normalise	A logical parameter to be used when the std_conc argument is defined. When normalise = TRUE the internal standard concentration is removed and the remaining phase concentrations normalised to sum to 100 percent.
tth_align	A vector defining the minimum and maximum 2theta values to be used during alignment (e.g. c(5,65)). 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.
shift	A single numeric value denoting the maximum (positive or negative) shift, in degrees 2theta, that is allowed during the shifting of selected phases. Default = 0.
tth_fps	A vector defining the minimum and maximum 2theta values to be used during automated full pattern summation (e.g. c(5,65)). If not defined, then the full range is used.
lod	Optional parameter used to define the limit of detection (in weight percent) of the internal standard (i.e. the phase provided in the std argument). The 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 lod = 0 then limits of detection are not computed.
amorphous	A character string of any phase ID's that should be treated as amorphous. These must match phases present in lib$phases$phase_id.
amorphous_lod	Optional parameter used to exclude amorphous phases if they are below this specified limit (percent). Must be between 0 and 100. Default = 0.
...	other arguments

Details

Applies automated full pattern summation 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 and harmonised)
`residuals`	a vector of the residuals (fitted vs measured)
`phases`	a dataframe of the phases used to produce the fitted pattern and their concentrations
`phases_grouped`	the phases dataframe grouped by phase_name and concentrations summed
`obj`	named vector of the objective parameters summarising the quality of the fit
`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

Bish, D.L., Post, J.E., 1989. Modern powder diffraction. Mineralogical Society of America.

Chipera, S.J., Bish, D.L., 2013. Fitting Full X-Ray Diffraction Patterns for Quantitative Analysis: A Method for Readily Quantifying Crystalline and Disordered Phases. Adv. Mater. Phys. Chem. 03, 47-53. doi:10.4236/ampc.2013.31A007

Chipera, S.J., Bish, D.L., 2002. FULLPAT: A full-pattern quantitative analysis program for X-ray powder diffraction using measured and calculated patterns. J. Appl. Crystallogr. 35, 744-749. doi:10.1107/S0021889802017405

Eberl, D.D., 2003. User's guide to RockJock - A program for determining quantitative mineralogy from powder X-ray diffraction data. Boulder, CA.

Examples

```
#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)

#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
`obj`	named vector of the objective parameters summarising the quality of the fit
`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

Chipera, S.J., Bish, D.L., 2013. Fitting Full X-Ray Diffraction Patterns for Quantitative Analysis: A Method for Readily Quantifying Crystalline and Disordered Phases. Adv. Mater. Phys. Chem. 03, 47-53. doi:10.4236/ampc.2013.31A007

Chipera, S.J., Bish, D.L., 2002. FULLPAT: A full-pattern quantitative analysis program for X-ray powder diffraction using measured and calculated patterns. J. Appl. Crystallogr. 35, 744-749. doi:10.1107/S0021889802017405

Eberl, D.D., 2003. User's guide to RockJock - A program for determining quantitative mineralogy from powder X-ray diffraction data. Boulder, CA.

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)
```

`fps.powdRlib` *Full pattern summation*

Description

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

Usage

```
## S3 method for class 'powdRlib'
fps(
  lib,
  smpl,
  harmonise,
  solver,
  obj,
  refs,
  std,
```

```
  force,
  std_conc,
  normalise,
  tth_align,
  align,
  manual_align,
  tth_fps,
  shift,
  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 using natural splines.
solver	The optimisation routine to be used. One of c("BFGS","Nelder-Mead","CG",or "NNLS"). Default = "BFGS".
obj	The objective function to minimise when "BFGS", "Nelder-Mead", or "CG" 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. If missing from the function call then all phases in the reference library will be used.
std	The phase ID (e.g. "QUA.1") to be used as internal standard. Must match an ID provided in the refs 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 lib$phases$phase_id or lib$phases$phase_name columns.
std_conc	The concentration of the internal standard (if known) in weight percent. If unknown then use std_conc = NA (default), in which case it will be assumed that all phases sum to 100 percent.
normalise	A logical parameter to be used when the std_conc argument is defined. When normalise = TRUE the internal standard concentration is removed and the remaining phase concentrations normalised to sum to 100 percent.
tth_align	A vector defining the minimum and maximum 2theta values to be used during alignment (e.g. c(5,65)). 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 (e.g. c(5,65)). If not defined, then the full range is used.
shift	A single numeric value denoting the maximum (positive or negative) shift, in degrees 2theta, that is allowed during the shifting of selected phases. Default = 0.
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.
...	other arguments

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 and their concentrations
phases_grouped	the phases dataframe grouped by phase_name and concentrations summed
obj	named vector of the objective parameters summarising the quality of the fit
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

Bish, D.L., Post, J.E., 1989. Modern powder diffraction. Mineralogical Society of America.

Chipera, S.J., Bish, D.L., 2013. Fitting Full X-Ray Diffraction Patterns for Quantitative Analysis: A Method for Readily Quantifying Crystalline and Disordered Phases. Adv. Mater. Phys. Chem. 03, 47-53. doi:10.4236/ampc.2013.31A007

Chipera, S.J., Bish, D.L., 2002. FULLPAT: A full-pattern quantitative analysis program for X-ray powder diffraction using measured and calculated patterns. J. Appl. Crystallogr. 35, 744-749. doi:10.1107/S0021889802017405

Eberl, D.D., 2003. User's guide to RockJock - A program for determining quantitative mineralogy from powder X-ray diffraction data. Boulder, CA.

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)
```

`minerals` *An example powdRlib reference library*

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

```
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

```
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.

`minerals_regroup_structure`
Example regrouping structure for the `minerals` *data*

Description

Example regrouping structure for the `minerals` data

Usage

```
minerals_regroup_structure
```

Format

A 2 column data frame.

First column contains the unique phase ID's of all phases in the `minerals` data. Second column contains the grouping structure for the data (Non-clay, Clay or Amorphous).

`minerals_xrd` *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, mode, xlim, 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.
`mode`	One of "fit", "residuals" or "both" defining whether to plot the fitted patterns, the residuals of the fit, or both, respectively. Default = "fit".

xlim	A numeric vector of length two providing limits of the x-axis. Defaults to full x-axis unless specified.
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

When seeking to inspect the results from full pattern summation, interactive plots are particularly useful and can be specified with the `interactive` argument.

Examples

```
#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

```
## 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

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)
```

---

plot.powdRfps *Plotting elements of a powdRfps object*

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, mode, xlim, 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.
mode	One of "fit", "residuals" or "both" defining whether to plot the fitted patterns, the residuals of the fit, or both, respectively. Default = "fit".
xlim	A numeric vector of length two providing limits of the x-axis. Defaults to full x-axis unless specified.
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

When seeking to inspect the results from full pattern summation, interactive plots are particularly useful and can be specified with the `interactive` argument.

Examples

```
#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

```
## 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

Plots can be made interactive using the logical `interactive` argument.

Examples

```
# 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)
```

---

powdR *powdR: Full Pattern Summation of X-Ray Powder Diffraction Data*

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

Chipera, S.J., Bish, D.L., 2013. Fitting Full X-Ray Diffraction Patterns for Quantitative Analysis: A Method for Readily Quantifying Crystalline and Disordered Phases. Adv. Mater. Phys. Chem. 03, 47-53. doi:10.4236/ampc.2013.31A007

Chipera, S.J., Bish, D.L., 2002. FULLPAT: A full-pattern quantitative analysis program for X-ray powder diffraction using measured and calculated patterns. J. Appl. Crystallogr. 35, 744-749. doi:10.1107/S0021889802017405

Eberl, D.D., 2003. User's guide to ROCKJOCK - A program for determining quantitative mineralogy from powder X-ray diffraction data. Boulder, CA.

powdRlib *Create an XRPD reference library*

Description

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

Usage

```
powdRlib(xrd_table, phases_table, check_names)
```

Arguments

xrd_table	A data frame of the count intensities of the XRPD reference patterns, all scaled to same maximum intensity, with their 2theta axis as the first column.
phases_table	A data frame of the required data (phase ID, phase name, and reference intensity ratio) for each reference pattern.
check_names	A logical argument defining whether the column names in the data supplied in xrd_table are syntactically valid variable names and are not duplicated. Default = TRUE.

Value

a list with components:

xrd	a data frame of reference patterns
tth	a vector of the 2theta axis
phases	a 3 column data frame of the IDs, names and reference intensity ratios of the reference pattern

Examples

```
#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)
```

regroup *regroup*

Description

regroup allows an alternative grouping structure to be applied to powdRfps and powdRafps objects. For more details see ?regroup.powdRfps or ?regroup.powdRafps.

Usage

```
regroup(x, ...)
```

Arguments

x	A powdRfps or powdRafps object
...	Other parameters passed to methods e.g. fps.powdRlib

Details

powdRfps and powdRafps objects contain a data frame called phases_grouped that summarises phase concentrations based on defined mineral groups from the powdRlib reference library. regroup allows you to change this grouping structure by supplying new group identities.

Value

a powdRfps or powdRafps object 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 regrouped according to the supplied data
rwp	the Rwp of the fitted vs measured pattern
weighted_pure_patterns	a data frame 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

Examples

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

#Load the soils data
data(soils)

#Load the regrouping structure
data(minerals_regroup_structure)

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

fps_sandstone_regrouped <- regroup(fps_sandstone,
                                   minerals_regroup_structure)

fps_sandstone_regrouped$phases_grouped


## End(Not run)
```

`regroup.powdRafps` *regroup*

Description

`regroup` allows an alternative grouping structure to be applied to `powdRfps` and `powdRafps` objects.

Usage

```
## S3 method for class 'powdRafps'
regroup(x, y, ...)
```

Arguments

`x`	A `powdRfps` or `powdRafps` object
`y`	A data frame. First column contains the phase IDs covering all those present in `x$phases$phase_id`. Second column contains the desired grouping of each phase.
`...`	other arguments

Details

`powdRfps` and `powdRafps` objects contain a data frame called `phases_grouped` that summarises phase concentrations based on defined mineral groups from the `powdRlib` reference library. `regroup` allows you to change this grouping structure by supplying new group identities.

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 and their concentrations
phases_grouped	the phases dataframe regrouped according to the supplied data
rwp	the Rwp of the fitted vs measured pattern
weighted_pure_patterns	a data frame 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

Examples

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

# Load the soils data
data(soils)

#Load the regrouping structure
data(minerals_regroup_structure)

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

afps_sandstone_regrouped <- regroup(afps_sandstone,
                                    minerals_regroup_structure)

afps_sandstone_regrouped$phases_grouped

## End(Not run)
```

regroup.powdRfps *regroup*

Description

regroup allows an alternative grouping structure to be applied to powdRfps and powdRafps objects.

Usage

```
## S3 method for class 'powdRfps'
regroup(x, y, ...)
```

Arguments

x	A powdRfps or powdRafps object
y	A data frame. First column contains the phase IDs covering all those present in x$phases$phase_id. Second column contains the desired grouping of each phase.
...	other arguments

Details

powdRfps and powdRafps objects contain a data frame called phases_grouped that summarises phase concentrations based on defined mineral groups from the powdRlib reference library. regroup allows you to change this grouping structure by supplying new group identities.

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 and their concentrations
phases_grouped	the phases dataframe regrouped according to the supplied data
rwp	the Rwp of the fitted vs measured pattern
weighted_pure_patterns	a data frame 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

Examples

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

#Load the soils data
data(soils)

#Load the regrouping structure
data(minerals_regroup_structure)

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

fps_sandstone_regrouped <- regroup(fps_sandstone,
                                   minerals_regroup_structure)

fps_sandstone_regrouped$phases_grouped


## End(Not run)
```

`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. Note that compared to same library supplied with RockJock the powdR patterns have been normalised to 10,000 counts and reference intensity ratios transformed so that all are relative to that of corundum, which has been set to a value of 1.0. 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)

Author(s)

Dennis Eberl

References

Eberl, D.D., 2003. User's guide to RockJock - A program for determining quantitative mineralogy from powder X-ray diffraction data. Boulder, CA.

---

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

Eberl, D.D., 2003. User's guide to RockJock - A program for determining quantitative mineralogy from powder X-ray diffraction data. Boulder, CA.

`run_powdR` *Run the powdR shiny app*

Description

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

Usage

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

Arguments

`...`	further arguments to pass to `runApp`

Examples

```
## 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

```
soils
```

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

`subset.powdRlib` *Subset a powdRlib object*

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

```
## 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

```
#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 a list of multiple powdRfps or powdRafps derived from fps() and afps(), respectively.

Usage

```
summarise_mineralogy(x, type, order, rwp, r, delta)
```

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 value as the final column in the output. This provides an objective measure of the difference between the fitted and measured patterns.
r	a logical operator denoting whether to include the R value as the final column in the output. This provides an objective measure of the difference between the fitted and measured patterns.
delta	a logical operator denoting whether to include the Delta value 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)

## 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)
```

`tth_transform` *Transform a two theta axis between wavelengths*

Description

`tth_transform` converts the two theta axis from one wavelength to another via Bragg's law. Use this function with caution if intending the apply `fps()` or `afps()` to wavelength transformed samples or libraries because background signals can vary with wavelength which may therefore affect the quality of the fit.

Usage

```
tth_transform(tth, from, to)
```

Arguments

`tth`	the 2theta vector to be transformed
`from`	numeric value defining the wavelength (Angstroms) to transform from
`to`	numeric value defining the wavelength (Angstroms) to transform to

Value

a transformed 2theta vector

Examples

```
data(soils)
sandstone2 <- soils$sandstone

#Convert from Cu (1.54056 Angstroms) to Co (1.78897 Angstroms)
sandstone2$tth <- tth_transform(sandstone2$tth,
                                from = 1.54056,
                                to = 1.78897)
#plot the change
plot(soils$sandstone, type = "l",
     xlim = c(0, 85))
lines(sandstone2, col = "red")

#Alternatively convert the 2theta axis of a library
data(minerals)

minerals2 <- minerals
minerals2$tth <- tth_transform(minerals2$tth,
                                from = 1.54056,
                                to = 1.78897)

#Plot the difference
plot(x = minerals$tth, y = minerals$xrd$QUA.1,
     type = "l", xlim = c(0, 85))
lines(x = minerals2$tth, y = minerals2$xrd$QUA.1,
      col = "red")
```

Index