Package ‘EMpeaksR’
December 20, 2021

Type     Package
Title    Conducting the Peak Fitting Based on the EM Algorithm
Version  0.2.0

Description The peak fitting of spectral data is performed by using the frame work of EM algo-
rithm. We adapted the EM algorithm for the peak fitting of spectral data set by considering the weight of the intensity corresponding to the measurement energy steps (Mat-
sumura, T., Nagamura, N., Akaho, S., Nagata, K., & Ando, Y. (2019 and 2021) <doi:10.1080/14686996.2019.1620123> and <doi:10.1080/27660400.2021.1899449>). The package efficiently estimates the parameters of Gaussian mixture model during iterative calculation between E-step and M-step, and the parameters are converged to a local optimal solution. This package can support the investigation of peak shift with two advantages: (1) a large amount of data can be processed at high speed; and (2) stable and automatic calculation can be easily performed.

License MIT + file LICENSE
Language en-US
Encoding UTF-8
RoxygenNote 7.1.1
NeedsCompilation no
Author Tarojiro Matsumura [aut, cre]
Maintainer Tarojiro Matsumura <matsumura-tarojiro@aist.go.jp>
Repository CRAN
Date/Publication 2021-12-20 14:02:04 UTC

R topics documented:

  show_dsgmm_curve .................................................. 2
  show_gmm_curve .................................................... 4
  show_lmm_curve .................................................... 6
  show_pvmm_curve ................................................... 7
  spect_em_dsgmm ................................................... 9
  spect_em_gmm ..................................................... 12
  spect_em_lmm ..................................................... 14
  spect_em_pvmm ................................................... 16
show_dsgmm_curve

Visualization of the result of spect_em_dsgmm

Description

Visualization of the result of spect_em_dsgmm().

Usage

show_dsgmm_curve(spect_em_dsgmm_res, x, y, mix_ratio_init, mu_init, sigma_init, alpha_init, eta_init)

Arguments

spect_em_dsgmm_res
data set obtained by spect_em_dsgmm()
x measurement steps
y intensity
mix_ratio_init initial values of the mixture ratio of the components
mu_init initial values of the mean of the components
sigma_init initial values of the standard deviation of the components
alpha_init initial values of the asymmetric parameter of the components
eta_init initial values of the mixing ratio of Gauss and Lorentz distribution

Details

Perform a visualization of fitting curve estimated by Doniach-Sunjic-Gauss mixture model.

Value

Show the fitting curve and variation of the parameters.

References


Examples

#generating the synthetic spectral data based on three component Doniach-Sunjic-Gauss mixture model.
x <- seq(0, 100, by = 0.5)
true_mu <- c(20, 50, 80)
true_sigma <- c(3, 3, 3)
true_alpha <- c(0.1, 0.3, 0.1)
true_eta <- c(0.4, 0.6, 0.1)
true_mix_ratio <- rep(1/3, 3)
degree <- 4

#trancated Doniach-Sunjic-Gauss
truncated_dsg <- function(x, mu, sigma, alpha, eta) {
  (((eta*(((gamma(1-alpha)) /
  ((x-mu)^2+(sqrt(2*log(2))*sigma)^2)^((1-alpha)/2)) *
  cos((pi*alpha/2)+(1-alpha)*atan((x-mu) /
  (sqrt(2*log(2))*sigma)))))) + (1-eta)*dnorm(x, mu, sigma)) /
  sum( ((eta*(((gamma(1-alpha)) /
  ((x-mu)^2+(sqrt(2*log(2))*sigma)^2)^((1-alpha)/2)) *
  cos((pi*alpha/2)+(1-alpha)*atan((x-mu) /
  (sqrt(2*log(2))*sigma)))))) + (1-eta)*dnorm(x, mu, sigma)))
}

y <- c(true_mix_ratio[1]*truncated_dsg(x = x,
  mu = true_mu[1],
  sigma = true_sigma[1],
  alpha = true_alpha[1],
  eta = true_eta[1]) * 10^degree +
true_mix_ratio[2]*truncated_dsg(x = x,
  mu = true_mu[2],
  sigma = true_sigma[2],
  alpha = true_alpha[2],
  eta = true_eta[2]) * 10^degree +
true_mix_ratio[3]*truncated_dsg(x = x,
  mu = true_mu[3],
  sigma = true_sigma[3],
  alpha = true_alpha[3],
  eta = true_eta[3]) * 10^degree)

plot(y~x, main = "genrated synthetic spectral data")

#Peak fitting by EMpeaksR
#Initial values
K <- 3
mix_ratio_init <- c(0.2, 0.4, 0.4)
mu_init <- c(20, 40, 70)
sigma_init <- c(4, 3, 2)
alpha_init <- c(0.3, 0.2, 0.4)
eta_init <- c(0.5, 0.4, 0.3)

#Coducting calculation
SP_ECM_DSG_res <- spect_em_dsgmm(x = x,
y = y,
show_gmm_curve

```r
mu = mu_init,
sigma = sigma_init,
alpha = alpha_init,
eta = eta_init,
mix_ratio = mix_ratio_init,
conv.cri = 1e-6,
maxit = 100)
```

# Plot fitting curve and trace plot of parameters
show_dsgmm_curve(SP_ECM_DSG_res,
                 x,
                 y,
                 mix_ratio_init,
                 mu_init,
                 sigma_init,
                 alpha_init,
                 eta_init)

# Showing the result of spect_em_dsgmm()
print(cbind(c(mu_init),
           c(sigma_init),
           c(alpha_init),
           c(eta_init),
           c(mix_ratio_init)))

print(cbind(SP_ECM_DSG_res$mu,
            SP_ECM_DSG_res$sigma,
            SP_ECM_DSG_res$alpha,
            SP_ECM_DSG_res$eta,
            SP_ECM_DSG_res$mix_ratio))

print(cbind(true_mu,
            true_sigma,
            true_alpha,
            true_eta,
            true_mix_ratio))
```

---

**show_gmm_curve**  
Visualization of the result of spect_em_gmm

**Description**

Visualization of the result of spect_em_gmm().

**Usage**

```r
show_gmm_curve(spect_em_gmm_res, x, y, mix_ratio_init, mu_init, sigma_init)
```
show_gmm_curve

Arguments

- `spect_em_gmm_res`: data set obtained by `spect_em_gmm()`
- `x`: measurement steps
- `y`: intensity
- `mix_ratio_init`: initial values of the mixture ratio of the components
- `mu_init`: initial values of the mean of the components
- `sigma_init`: initial values of the standard deviation of the components

Details

Perform a visualization of fitting curve estimated by Gaussian mixture model.

Value

Show the fitting curve and variation of the parameters.

References


Examples

```r
# generating the synthetic spectral data based on three component Gaussian mixture model.
x <- seq(0, 100, by = 0.5)
true_mu <- c(35, 50, 65)
true_sigma <- c(3, 3, 3)
ture_mix_ratio <- rep(1/3, 3)
degree <- 4
y <- c(true_mix_ratio[1] * dnorm(x = x, mean = true_mu[1], sd = true_sigma[1]) * 10^degree +
true_mix_ratio[2] * dnorm(x = x, mean = true_mu[2], sd = true_sigma[2]) * 10^degree +
ture_mix_ratio[3] * dnorm(x = x, mean = true_mu[3], sd = true_sigma[3]) * 10^degree)
plot(y~x, main = "generated synthetic spectral data")

# Peak fitting by EMpeaksR
# Initial values
K <- 3
mix_ratio_init <- c(0.2, 0.4, 0.4)
mu_init <- c(20, 40, 70)
sigma_init <- c(2, 5, 4)

# Conducting calculation
SP_EM_G_res <- spect_em_gmm(x, y, mu = mu_init, sigma = sigma_init, mix_ratio = mix_ratio_init,
conv.cri = 1e-8, maxit = 100000)
```
# Plot fitting curve and trace plot of parameters
show_gmm_curve(SP_EM_G_res, x, y, mix_ratio_init, mu_init, sigma_init)

# Showing the result of spect_em_gmm()
print(cbind(c(mu_init), c(sigma_init), c(mix_ratio_init)))
print(cbind(SP_EM_G_res$mu, SP_EM_G_res$sigma, SP_EM_G_res$mix_ratio))
print(cbind(true_mu, true_sigma, true_mix_ratio))

Description
Visualization of the result of spect_em_lmm().

Usage
show_lmm_curve(spect_em_lmm_res, x, y, mix_ratio_init, mu_init, gam_init)

Arguments
spect_em_lmm_res
data set obtained by spect_em_lmm()
x
measurement steps
y
intensity
mix_ratio_init
initial values of the mixture ratio of the components
mu_init
initial values of the mean of the components
gam_init
initial values of the scale parameter of the components

Details
Perform a visualization of fitting curve estimated by Lorentz mixture model.

Value
Show the fitting curve and variation of the parameters.

References
show_pvmm_curve

Examples

# Generating the synthetic spectral data based on three component Lorentz mixture model.
x <- seq(0, 100, by = 0.5)
true_mu <- c(35, 50, 65)
true_gam <- c(3, 3, 3)
true_mix_ratio <- rep(1/3, 3)
degree <- 4

# Normalized Lorentz distribution
dCauchy <- function(x, mu, gam) {
  (dcauchy(x, mu, gam)) / sum(dcauchy(x, mu, gam))
}
y <- c(true_mix_ratio[1] * dCauchy(x = x, mu = true_mu[1], gam = true_gam[1]) * 10^degree +
      true_mix_ratio[2] * dCauchy(x = x, mu = true_mu[2], gam = true_gam[2]) * 10^degree +
      true_mix_ratio[3] * dCauchy(x = x, mu = true_mu[3], gam = true_gam[3]) * 10^degree)

plot(y~x, main = "generated synthetic spectral data")

# Peak fitting by EMpeaksR
# Initial values
K <- 3
mix_ratio_init <- c(0.2, 0.4, 0.4)
mu_init <- c(20, 40, 70)
gam_init <- c(2, 5, 4)

# Conducting calculation
SP_ECM_L_res <- spect_em_lmm(x, y, mu = mu_init, gam = gam_init, mix_ratio = mix_ratio_init,
  conv.cri = 1e-6, maxit = 100000)

# Plotting fitting curve and trace plot of parameters
show_lmm_curve(SP_ECM_L_res, x, y, mix_ratio_init, mu_init, gam_init)

# Showing the result of spect_em_lmm()
print(cbind(c(mu_init), c(gam_init), c(mix_ratio_init)))
print(cbind(SP_ECM_L_res$mu, SP_ECM_L_res$gam, SP_ECM_L_res$mix_ratio))
print(cbind(true_mu, true_gam, true_mix_ratio))

show_pvmm_curve

Visualization of the result of spect_em_pvmm

Description

Visualization of the result of spect_em_pvmm().

Usage

show_pvmm_curve(spect_em_pvmm_res, x, y, mix_ratio_init, mu_init, sigma_init, eta_init)
Arguments

- `spect_em_pvmm_res`: data set obtained by `spect_em_pvmm()`
- `x`: measurement steps
- `y`: intensity
- `mix_ratio_init`: initial values of the mixture ratio of the components
- `mu_init`: initial values of the mean of the components
- `sigma_init`: initial values of the standard deviation of the components
- `eta_init`: initial values of the mixing ratio of Gauss and Lorentz distribution

Details

Perform a visualization of fitting curve estimated by Pseudo-Voigt mixture model.

Value

Show the fitting curve and variation of the parameters.

References


Examples

```r
#generating the synthetic spectral data based on three component Pseudo-Voigt mixture model.
x <- seq(0, 100, by = 0.5)
true_mu <- c(35, 50, 65)
true_sigma <- c(3, 3, 3)
true_eta <- c(0.3, 0.8, 0.5)
true_mix_ratio <- rep(1/3, 3)
degree <- 4

#Normalized Pseudo-Voigt distribution
truncated_pv <- function(x, mu, sigma, eta) {
  (eta*dcauchy(x, mu, sqrt(2*log(2))*sigma) + (1-eta)*dnorm(x, mu, sigma)) / 
  sum(eta*dcauchy(x, mu, sqrt(2*log(2))*sigma) + (1-eta)*dnorm(x, mu, sigma))
}

y <- c(true_mix_ratio[1]*truncated_pv(x = x, 
  mu = true_mu[1], 
  sigma = true_sigma[1], 
  eta = true_eta[1])
  + true_mix_ratio[2]*truncated_pv(x = x, 
  mu = true_mu[2], 
  sigma = true_sigma[2], 
  eta = true_eta[2])
  + 
  true_mix_ratio[3]*truncated_pv(x = x, 
  mu = true_mu[3], 
  sigma = true_sigma[3], 
  eta = true_eta[3])
  + 
  10^degree
)
```
spec_em_dsgmm

\[
\text{mu} = \text{true}_\text{mu}[2], \\
\text{sigma} = \text{true}_\text{sigma}[2], \\
\text{eta} = \text{true}_\text{eta}[2] \times 10^{\text{degree}} + \\
\text{true}_\text{mix}_\text{ratio}[3] \times \text{truncated pv}(x = x, \\
\text{mu} = \text{true}_\text{mu}[3], \\
\text{sigma} = \text{true}_\text{sigma}[3], \\
\text{eta} = \text{true}_\text{eta}[3] \times 10^{\text{degree}})
\]

\[
\text{plot}(y \sim x, \text{main} = \text{"generated synthetic spectral data"})
\]

# Peak fitting by EMpeaksR
# Initial values
K <- 3

mix_ratio_init <- c(0.2, 0.4, 0.4) \\
mu_init <- c(20, 40, 70) \\
sigma_init <- c(2, 5, 4) \\
eta_init <- c(0.5, 0.4, 0.3)

# Conducting calculation
SP_ECM_PV_res <- spec_em_pvmm(x = x, \\
y = y, \\
mu = mu_init, \\
sigma = sigma_init, \\
eta = eta_init, \\
mix_ratio = mix_ratio_init, \\
conv.cri = 1e-6, \\
maxit = 100000)

# Plot fitting curve and trace plot of parameters
show_pvmm_curve(SP_ECM_PV_res, x, y, mix_ratio_init, mu_init, sigma_init, eta_init)

# Showing the result of spec_em_pvmm()
print(cbind(c(mu_init), c(sigma_init), c(eta_init), c(mix_ratio_init)))
print(cbind(SP_ECM_PV_res$mu, SP_ECM_PV_res$sigma, SP_ECM_PV_res$eta, SP_ECM_PV_res$mix_ratio))
print(cbind(true_mu, true_sigma, true_eta, true_mix_ratio))

---

**spect_em_dsgmm**  
_Spectrum adapted ECM algorithm by DSGMM_

**Description**

Perform a peak fitting based on the spectrum adapted ECM algorithm by Doniach-Sunjic-Gauss mixture model.

**Usage**

```
spect_em_dsgmm(x, y, mu, sigma, alpha, eta, mix_ratio, conv.cri, maxit)
```
Arguments

- x: measurement steps
- y: intensity
- mu: mean of the components
- sigma: standard deviation of the components
- alpha: asymmetric parameter of the component
- eta: mixing ratio of Gauss and Lorentz distribution
- mix_ratio: mixture ratio of the components
- conv.cri: criterion of the convergence
- maxit: maximum number of the iteration

Details

Peak fitting is conducted by spectrum adapted ECM algorithm.

Value

- mu: estimated mean of the components
- sigma: estimated standard deviation of the components
- alpha: estimated asymmetric parameter of the components
- eta: estimated mixing ratio of Gauss and Lorentz distribution
- mix_ratio: estimated mixture ratio of the components
- it: number of the iteration to reach the convergence
- LL: variation of the weighted log likelihood values
- MU: variation of mu
- SIGMA: variation of sigma
- ALPHA: variation of alpha
- ETA: variation of beta
- MIX_RATIO: variation of mix_ratio
- W_K: decomposed component of the spectral data
- convergence: message for the convergence in the calculation
- cal_time: calculation time to complete the peak fitting. Unit is seconds

References


Examples

# generating the synthetic spectral data based on three component Doniach-Sunjic-Gauss mixture model.

```r
x <- seq(0, 100, by = 0.5)
true_mu <- c(20, 50, 80)
true_sigma <- c(3, 3, 3)
true_alpha <- c(0.1, 0.3, 0.1)
true_eta <- c(0.4, 0.6, 0.1)
true_mix_ratio <- rep(1/3, 3)
degree <- 4

# truncated Doniach-Sunjic-Gauss
truncated_dsg <- function(x, mu, sigma, alpha, eta) {
  ((eta*((gamma(1-alpha))/((x-mu)^2+(sqrt(2*log(2))*sigma)^2)^((1-alpha)/2)) *
     cos((pi*alpha/2)+(1-alpha)*atan((x-mu)/(sqrt(2*log(2))*sigma)))) + (1-eta)*dnorm(x, mu, sigma)) /
  sum( ((eta*((gamma(1-alpha))/((x-mu)^2+(sqrt(2*log(2))*sigma)^2)^((1-alpha)/2)) *
     cos((pi*alpha/2)+(1-alpha)*atan((x-mu)/(sqrt(2*log(2))*sigma)))) + (1-eta)*dnorm(x, mu, sigma)))
}

y <- c(true_mix_ratio[1]*truncated_dsg(x = x, 
         mu = true_mu[1],
         sigma = true_sigma[1],
         alpha = true_alpha[1],
         eta = true_eta[1]) * 10^degree +

true_mix_ratio[2]*truncated_dsg(x = x, 
         mu = true_mu[2],
         sigma = true_sigma[2],
         alpha = true_alpha[2],
         eta = true_eta[2]) * 10^degree +

true_mix_ratio[3]*truncated_dsg(x = x, 
         mu = true_mu[3],
         sigma = true_sigma[3],
         alpha = true_alpha[3],
         eta = true_eta[3]) * 10^degree)

plot(y ~ x, main = "generated synthetic spectral data")
```

# Peak fitting by EMpeaksR

# Initial values

```r
K <- 3
mix_ratio_init <- c(0.2, 0.4, 0.4)
mu_init <- c(20, 40, 70)
sigma_init <- c(4, 3, 2)
alpha_init <- c(0.3, 0.2, 0.4)
eta_init <- c(0.5, 0.4, 0.3)
```

# Conducting calculation

```r
SP_ECM_DSG_res <- spect_em_dsgmm(x = x, 
                                 y = y, 
                                 K = K, 
                                 mix_ratio_init = mix_ratio_init, 
                                 mu_init = mu_init, 
                                 sigma_init = sigma_init, 
                                 alpha_init = alpha_init, 
                                 eta_init = eta_init)
```
spect_em_gmm

Spectrum adapted EM algorithm by GMM

Description

Perform a peak fitting based on the spectrum adapted EM algorithm by Gaussian mixture model.

Usage

spect_em_gmm(x, y, mu, sigma, mix_ratio, conv.cri, maxit)
Arguments

x  measurement steps
y  intensity
mu mean of the components
sigma standard deviation of the components
mix_ratio mixture ratio of the components
conv.cri criterion of the convergence
maxit maximum number of the iteration

Details

Peak fitting is conducted by spectrum adapted EM algorithm.

Value

mu estimated mean of the components
sigma estimated standard deviation of the components
mix_ratio estimated mixture ratio of the components
it number of the iteration to reach the convergence
LL variation of the weighted log likelihood values
MU variation of mu
SIGMA variation of sigma
MIX_RATIO variation of mix_ratio
W_K decomposed component of the spectral data
convergence message for the convergence in the calculation
cal_time calculation time to complete the peak fitting. Unit is seconds

References


Examples

#generating the synthetic spectral data based on three component Gaussian mixture model.
x <- seq(0, 100, by = 0.5)
true_mu <- c(35, 50, 65)
true_sigma <- c(3, 3, 3)
true_mix_ratio <- rep(1/3, 3)
degree <- 4

y <- c(true_mix_ratio[1] * dnorm(x = x, mean = true_mu[1], sd = true_sigma[1]))*10*degree +
       true_mix_ratio[2] * dnorm(x = x, mean = true_mu[2], sd = true_sigma[2]))*10*degree +
       true_mix_ratio[3] * dnorm(x = x, mean = true_mu[3], sd = true_sigma[3]))*10*degree)
plot(y~x, main = "generated synthetic spectral data")

# Peak fitting by EMpeaksR
# Initial values
K <- 3

mix_ratio_init <- c(0.2, 0.4, 0.4)
mu_init <- c(20, 40, 70)
sigma_init <- c(2, 5, 4)

# Conducting calculation
SP_EM_G_res <- spect_em_gmm(x, y, mu = mu_init, sigma = sigma_init, mix_ratio = mix_ratio_init,
conv.cri = 1e-8, maxit = 100000)

# Plot fitting curve and trace plot of parameters
show_gmm_curve(SP_EM_G_res, x, y, mix_ratio_init, mu_init, sigma_init)

# Showing the result of spect_em_gmm()
print(cbind(c(mu_init), c(sigma_init), c(mix_ratio_init)))
print(cbind(SP_EM_G_res$mu, SP_EM_G_res$sigma, SP_EM_G_res$mix_ratio))
print(cbind(true_mu, true_sigma, true_mix_ratio))

---

spect_em_lmm  

Spectrum adapted ECM algorithm by LMM

Description
Perform a peak fitting based on the spectrum adapted ECM algorithm by Lorentz mixture model.

Usage
spect_em_lmm(x, y, mu, gam, mix_ratio, conv.cri, maxit)

Arguments
- x: measurement steps
- y: intensity
- mu: mean of the components
- gam: scale parameter of the components
- mix_ratio: mixture ratio of the components
- conv.cri: criterion of the convergence
- maxit: maximum number of the iteration

Details
Peak fitting is conducted by spectrum adapted ECM algorithm.
Value

- **mu**: estimated mean of the components
- **gam**: estimated scale parameter of the components
- **mix_ratio**: estimated mixture ratio of the components
- **it**: number of the iteration to reach the convergence
- **LL**: variation of the weighted log likelihood values
- **MU**: variation of mu
- **GAM**: variation of gam
- **MIX_RATIO**: variation of mix_ratio
- **W_K**: decomposed component of the spectral data
- **convergence**: message for the convergence in the calculation
- **cal_time**: calculation time to complete the peak fitting. Unit is seconds

References


Examples

```r
#generating the synthetic spectral data based on three component Lorentz mixture model.
x <- seq(0, 100, by = 0.5)
true_mu <- c(35, 50, 65)
true_gam <- c(3, 3, 3)
true_mix_ratio <- rep(1/3, 3)
degree <- 4

#Normalized Lorentz distribution
dCauchy <- function(x, mu, gam) {
  (dcauchy(x, mu, gam)) / sum(dcauchy(x, mu, gam))
}

y <- c(true_mix_ratio[1] * dCauchy(x = x, mu = true_mu[1], gam = true_gam[1]) * 10^degree +
  true_mix_ratio[2] * dCauchy(x = x, mu = true_mu[2], gam = true_gam[2]) * 10^degree +
  true_mix_ratio[3] * dCauchy(x = x, mu = true_mu[3], gam = true_gam[3]) * 10^degree)

plot(y~x, main = "generated synthetic spectral data")

#Peak fitting by EMpeaksR
#Initial values
K <- 3
mix_ratio_init <- c(0.2, 0.4, 0.4)
```
mu_init <- c(20, 40, 70)
gam_init <- c(2, 5, 4)

# Conducting calculation
SP_ECM_L_res <- spect_em_lmm(x, y, mu = mu_init, gam = gam_init, mix_ratio = mix_ratio_init,
                              conv.cri = 1e-6, maxit = 100000)

# Plot fitting curve and trace plot of parameters
show_lmm_curve(SP_ECM_L_res, x, y, mix_ratio_init, mu_init, gam_init)

# Showing the result of spect_em_lmm()
print(cbind(c(mu_init), c(gam_init), c(mix_ratio_init)))
print(cbind(SP_ECM_L_res$mu, SP_ECM_L_res$gam, SP_ECM_L_res$mix_ratio))
print(cbind(true_mu, true_gam, true_mix_ratio))

---

spect_em_pvmm  Spectrum adapted ECM algorithm by PVMM

Description

Perform a peak fitting based on the spectrum adapted ECM algorithm by Pseudo-Voigt mixture model.

Usage

spect_em_pvmm(x, y, mu, sigma, eta, mix_ratio, conv.cri, maxit)

Arguments

- x  measurement steps
- y  intensity
- mu mean of the components
- sigma standard deviation of the components
- eta  mixing ratio of Gauss and Lorentz distribution
- mix_ratio  mixture ratio of the components
- conv.cri criterion of the convergence
- maxit maximum number of the iteration

Details

Peak fitting is conducted by spectrum adapted ECM algorithm.
Value

- **mu** estimated mean of the components
- **sigma** estimated standard deviation of the components
- **eta** estimated mixing ratio of Gauss and Lorentz distribution
- **mix_ratio** estimated mixture ratio of the components
- **it** number of the iteration to reach the convergence
- **LL** variation of the weighted log likelihood values
- **MU** variation of mu
- **SIGMA** variation of sigma
- **ETA** variation of beta
- **MIX_RATIO** variation of mix_ratio
- **W_K** decomposed component of the spectral data
- **convergence** message for the convergence in the calculation
- **cal_time** calculation time to complete the peak fitting. Unit is seconds

References


Examples

```r
#generating the synthetic spectral data based on three component Pseudo-Voigt mixture model.
x <- seq(0, 100, by = 0.5)
true_mu <- c(35, 50, 65)
true_sigma <- c(3, 3, 3)
true_eta <- c(0.3, 0.8, 0.5)
true_mix_ratio <- rep(1/3, 3)
degree <- 4

#Normalized Pseudo-Voigt distribution
truncated_pv <- function(x, mu, sigma, eta) {
  (eta*dcauchy(x, mu, sqrt(2*log(2))*sigma) + (1-eta)*dnorm(x, mu, sigma)) / 
  sum(eta*dcauchy(x, mu, sqrt(2*log(2))*sigma) + (1-eta)*dnorm(x, mu, sigma))
}

y <- c(true_mix_ratio[1]*truncated_pv(x = x, 
  mu = true_mu[1],
  sigma = true_sigma[1],
  eta = true_eta[1]) * 10^degree +
  true_mix_ratio[2]*truncated_pv(x = x, 
  mu = true_mu[2],
  sigma = true_sigma[2],
  eta = true_eta[2]) * 10^degree +
  true_mix_ratio[3]*truncated_pv(x = x, 
  mu = true_mu[3],
  sigma = true_sigma[3],
  eta = true_eta[3]) * 10^degree)
```
\[
\text{sigma} = \text{true_sigma}[2], \\
\text{eta} = \text{true_eta}[2] \times 10^{\text{degree}} + \\
\text{true_mix_ratio}[3] \times \text{truncated_pv}(x = x, \\
\quad \mu = \text{true_mu}[3], \\
\quad \sigma = \text{true_sigma}[3], \\
\quad \eta = \text{true_eta}[3] \times 10^{\text{degree}})
\]

\text{plot}(y \sim x, \text{main} = \text{"generated synthetic spectral data"})

# Peak fitting by EMpeaksR
# Initial values
K <- 3

mix_ratio_init <- c(0.2, 0.4, 0.4)
mu_init <- c(20, 40, 70)
sigma_init <- c(2, 5, 4)
eta_init <- c(0.5, 0.4, 0.3)

# Conducting calculation
SP_ECM_PV_res <- spect_em_pvmm(x = x, 
\quad y = y, 
\quad mu = mu_init, 
\quad sigma = sigma_init, 
\quad eta = eta_init, 
\quad mix_ratio = mix_ratio_init, 
\quad conv.cri = 1e-6, 
\quad maxit = 100000)

# Plot fitting curve and trace plot of parameters
show_pvmm_curve(SP_ECM_PV_res, x, y, mix_ratio_init, mu_init, sigma_init, eta_init)

# Showing the result of spect_em_pvmm()
print(cbind(c(mu_init), c(sigma_init), c(eta_init), c(mix_ratio_init)))
print(cbind(SP_ECM_PV_res$mu, SP_ECM_PV_res$sigma, SP_ECM_PV_res$eta, SP_ECM_PV_res$mix_ratio))
print(cbind(true_mu, true_sigma, true_eta, true_mix_ratio))
Index

show_dsgmm_curve, 2
show_gmm_curve, 4
show_lmm_curve, 6
show_pvmm_curve, 7
spect_em_dsgmm, 9
spect_em_gmm, 12
spect_em_lmm, 14
spect_em_pvmm, 16