Package ‘EMpeaksR’

March 29, 2023

Type Package
Title Conducting the Peak Fitting Based on the EM Algorithm
Version 0.3.1
Description The peak fitting of spectral data is performed by using the framework of EM algorithm. 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 (Matsumura, T., Nagamura, N., Akaho, S., Nagata, K., & Ando, Y. (2019, 2021 and 2023) <doi:10.1080/14686996.2019.1620123>, <doi:10.1080/27660400.2021.1899449>, <doi:10.1080/27660400.2022.2159753>). 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
Suggests testthat (>= 3.0.0)

Author Tarojiro Matsumura [aut, cre]
Maintainer Tarojiro Matsumura <matsumura-tarojiro@aist.go.jp>
Repository CRAN
Date/Publication 2023-03-29 13:40:02 UTC

R topics documented:

  show_dsgmm_curve .................................................. 2
  show_gmm_curve .................................................. 4
  show_lmm_curve .................................................. 6
  show_pvmm_curve .................................................. 8
**show_dsgmm_curve**

Visualization of the result of `spect_em_dsgmm()`.

**Usage**

```r
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)

# Conducting calculation
SP_ECM_DSG_res <- spect_em_dsgmm(x = x,
y = y,
mu = mu_init,
sigma = sigma_init,
alpha = alpha_init,
eta = eta_init,
mix_ratio = mix_ratio_init,
conv.cri = 1e-2,
maxit = 2000)

# 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

Description
Visualization of the result of spect_em_gmm().

Usage
show_gmm_curve(spect_em_gmm_res, x, y, mix_ratio_init, mu_init, sigma_init)

Arguments
spect_em_gmm_res
data set obtained by spect_em_gmm()
xmeasurement steps
yintensity
mix_ratio_initinitial values of the mixture ratio of the components
mu_initinitial values of the mean of the components
sigma_initinitial 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
#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-2, maxit = 2000)

# Plotting 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))

show_lmm_curve <- Visualization of the result of spect_em_lmm

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

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()
xmeasurement steps
yy
mix_ratio_initinitial values of the mixture ratio of the components
mu_initinitial values of the mean of the components
gam_initinitial 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


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)
dergee <- 4

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-2, maxit = 2000)

# 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))
Visualization of the result of spect_em_pvmm

Description

Visualization of the result of spect_em_pvmm().

Usage

\[
\text{show_pvmm_curve(spect_em_pvmm_res, x, y, mix_ratio_init, mu_init, sigma_init, eta_init)}
\]

Arguments

- \text{spect_em_pvmm_res}  
  data set obtained by spect_em_pvmm()
- \text{x}  
  measurement steps
- \text{y}  
  intensity
- \text{mix_ratio_init}  
  initial values of the mixture ratio of the components
- \text{mu_init}  
  initial values of the mean of the components
- \text{sigma_init}  
  initial values of the standard deviation of the components
- \text{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])*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)

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)
eta_init <- c(0.5, 0.4, 0.3)

# Conducting calculation

SP_ECM_PV_res <- spect_em_pvmm(x = x, y = y, 
                               mu = mu_init, 
                               sigma = sigma_init, 
                               eta = eta_init, 
                               mix_ratio = mix_ratio_init, 
                               conv.cri = 1e-2, 
                               maxit = 2000)

# 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))

show_pvmm_lback_curve

Description
Visualization of the result of spect_em_pvmm_lback().

Usage
show_pvmm_lback_curve(spect_em_pvmm_lback_res, x, y, mix_ratio_init, mu_init, sigma_init, eta_init, x_lower, x_upper)

Arguments
spect_em_pvmm_lback_res
data set obtained by spect_em_pvmm_lback()
x measurement steps
y intensity
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
mix_ratio_init initial values of the mixture ratio of the components
x_lower lower limit of the measurement steps. Default is a minimum of x
x_upper upper limit of the measurement steps. Default is a maximum of x

Details
Perform a visualization of fitting curve estimated by pseudo-Voigt mixture model with a linear background.

Value
Show the fitting curve and variation of the parameters.
References

Examples
#generating the synthetic spectral data based on three component Pseudo-Voigt mixture model.
x <- seq(0, 100, by = 0.5)
K <- 3
true_mu <- c(35, 50, 65)
true_sigma <- c(3, 3, 3)
true_mix_ratio <- c(0.5/3, 0.5/3, 0.5/3, 0.5)
true_eta <- c(0.4, 0.6, 0.1)
dergee <- 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])^degree +
true_mix_ratio[2]*truncated_pv(x = x, 
  mu = true_mu[2],
  sigma = true_sigma[2],
  eta = true_eta[2])^degree +
true_mix_ratio[3]*truncated_pv(x = x, 
  mu = true_mu[3],
  sigma = true_sigma[3],
  eta = true_eta[3])^degree +
true_mix_ratio[4]*(c(500*x + 15000) / sum(500*x + 15000))^degree)

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

#Peak fitting by EMpeaksR
#Initial values
mu_init <- c(30, 40, 60)
sigma_init <- c(4, 4, 4)
mix_ratio_init <- rep(1/(length(mu_init)+1), length(mu_init)+1)
eta_init <- c(1, 1, 1)
# Conducting calculation
```
SP_ECM_PV_LBACK_res <- spect_em_pvmm_lback(x = x, 
    y = y, 
    mu = mu_init, 
    sigma = sigma_init, 
    eta = eta_init, 
    mix_ratio = mix_ratio_init, 
    x_lower = min(x), 
    x_upper = max(x), 
    conv.cri = 1e-2, 
    maxit = 2000)
```

# Plot fitting curve and trace plot of parameters
```
show_pvmm_lback_curve(spect_em_pvmm_lback_res = SP_ECM_PV_LBACK_res, 
    x = x, 
    y = y, 
    mix_ratio_init = mix_ratio_init, 
    mu_init = mu_init, 
    sigma_init = sigma_init, 
    eta_init = eta_init, 
    x_lower = min(x), 
    x_upper = max(x))
```

# Showing the result of spect_em_pvmm_lback()
```
print(cbind(SP_ECM_PV_LBACK_res$mu, SP_ECM_PV_LBACK_res$sigma, SP_ECM_PV_LBACK_res$eta, 
    SP_ECM_PV_LBACK_res$mix_ratio[1:K]))
print(cbind(true_mu, true_sigma, true_eta, true_mix_ratio[1:K]))
```

---

**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
### Details

Peak fitting is conducted by spectrum adapted ECM algorithm.

### Value

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

### 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

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
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, 0.3, 0.4)
eta_init <- c(0.5, 0.4, 0.3)

#Conducting calculation
SP_ECM_DSG_res <- spect_em_dsgmm(x = x, 
y = y,
spect_em_gmm

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

```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)
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-2, maxit = 2000)

# 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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mu</td>
<td>estimated mean of the components</td>
</tr>
<tr>
<td>gam</td>
<td>estimated scale parameter of the components</td>
</tr>
<tr>
<td>mix_ratio</td>
<td>estimated mixture ratio of the components</td>
</tr>
<tr>
<td>it</td>
<td>number of the iteration to reach the convergence</td>
</tr>
<tr>
<td>LL</td>
<td>variation of the weighted log likelihood values</td>
</tr>
<tr>
<td>MU</td>
<td>variation of mu</td>
</tr>
<tr>
<td>GAM</td>
<td>variation of gam</td>
</tr>
<tr>
<td>MIX_RATIO</td>
<td>variation of mix_ratio</td>
</tr>
<tr>
<td>W_K</td>
<td>decomposed component of the spectral data</td>
</tr>
<tr>
<td>convergence</td>
<td>message for the convergence in the calculation</td>
</tr>
<tr>
<td>cal_time</td>
<td>calculation time to complete the peak fitting. Unit is seconds</td>
</tr>
</tbody>
</table>

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)
```
**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

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mu</td>
<td>estimated mean of the components</td>
</tr>
<tr>
<td>sigma</td>
<td>estimated standard deviation of the components</td>
</tr>
<tr>
<td>eta</td>
<td>estimated mixing ratio of Gauss and Lorentz distribution</td>
</tr>
<tr>
<td>mix_ratio</td>
<td>estimated mixture ratio of the components</td>
</tr>
<tr>
<td>it</td>
<td>number of the iteration to reach the convergence</td>
</tr>
<tr>
<td>LL</td>
<td>variation of the weighted log likelihood values</td>
</tr>
<tr>
<td>MU</td>
<td>variation of mu</td>
</tr>
<tr>
<td>SIGMA</td>
<td>variation of sigma</td>
</tr>
<tr>
<td>ETA</td>
<td>variation of beta</td>
</tr>
<tr>
<td>MIX_RATIO</td>
<td>variation of mix_ratio</td>
</tr>
<tr>
<td>W_K</td>
<td>decomposed component of the spectral data</td>
</tr>
</tbody>
</table>

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])
)```
```R
spect_em_pvmm_lback

sigma = true_sigma[2],
etta = true_eta[2])*10^degree +
true_mix_ratio[3]*truncated_pv(x = x,
mu = true_mu[3],
sigma = true_sigma[3],
etta = true_eta[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)
etta_init <- c(0.5, 0.4, 0.3)

#Conducting calculation
SP_ECM_PV_res <- spect_em_pvmm_lback(x = x,
y = y,
mu = mu_init,
sigma = sigma_init,
etta = etta_init,
mix_ratio = mix_ratio_init,
conv.cri = 1e-2,
maxit = 2000)

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

#Showing the result of spect_em_pvmm()
print(cbind(c(mu_init), c(sigma_init), c(etta_init), c(mix_ratio_init)))
print(cbind(SP_ECM_PV_res$mu, SP_ECM_PV_res$sigma, SP_ECM_PV_res$etta, SP_ECM_PV_res$mix_ratio))
print(cbind(true_mu, true_sigma, true_etta, true_mix_ratio))
```

---

**spect_em_pvmm_lback**

*Spectrum adapted ECM algorithm by PVMM with a linear background*

**Description**

Perform a peak fitting based on the spectrum adapted ECM algorithm by pseudo-Voigt mixture model with a linear background.

**Usage**

```R
spect_em_pvmm_lback(x, y, mu, sigma, etta, mix_ratio, x_lower, x_upper, 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
- `x_lower`: lower limit of the measurement steps. Default is a minimum of `x`
- `x_upper`: upper limit of the measurement steps. Default is a maximum of `x`
- `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 `eta`
- `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 Pseudo-Voigt mixture model.
x <- seq(0, 100, by = 0.5)
K <- 3
true_mu <- c(35, 50, 65)
true_sigma <- c(3, 3, 3)
true_mix_ratio <- c(0.5/3, 0.5/3, 0.5/3, 0.5)
true_eta <- c(0.4, 0.6, 0.1)
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 +
true_mix_ratio[4]*(c(500*x + 15000) / sum(500*x + 15000)) * 10^degree)

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

# Peak fitting by EMpeaksR
# Initial values
mu_init <- c(30, 40, 60)
sigma_init <- c(4, 4, 4)
mix_ratio_init <- rep(1/(length(mu_init)+3), length(mu_init)+3)
eta_init <- c(1, 1, 1)

# Conducting calculation
SP_ECM_PV_LBACK_res <- spect_em_pvmm_lback(x = x, y = y, mu = mu_init, sigma = sigma_init, eta = eta_init, mix_ratio = mix_ratio_init, x_lower = min(x), x_upper = max(x), conv.cri = 1e-2, maxit = 2000)

# Plot fitting curve and trace plot of parameters
show_pvmm_lback_curve(spect_em_pvmm_lback_res = SP_ECM_PV_LBACK_res,
    x = x,
    y = y,
    mix_ratio_init = mix_ratio_init,
    mu_init = mu_init,
    sigma_init = sigma_init,
    eta_init = eta_init,
    x_lower = min(x),
    x_upper = max(x))

#Showing the result of spect_em_pvmm_lback()
print(cbind(SP_ECM_PV_LBACK_res$mu, SP_ECM_PV_LBACK_res$sigma, SP_ECM_PV_LBACK_res$eta,
             SP_ECM_PV_LBACK_res$mix_ratio[1:K]))

print(cbind(true_mu, true_sigma, true_eta, true_mix_ratio[1:K]))
Index

show_dsgmm_curve, 2
show_gmm_curve, 4
show_lmm_curve, 6
show_pvmm_curve, 8
show_pvmm_lback_curve, 10
spect_em_dsgmm, 12
spect_em_gmm, 15
spect_em_lmm, 17
spect_em_pvmm, 19
spect_em_pvmm_lback, 21