

# Package ‘baselineARPLss’

September 23, 2024

**Type** Package

**Title** Baseline correction with arPLS smoothing (Baek 2015)

**Version** 0.1.0

**Description** Implements the algorithm for smoothing of spectra from: Sung-June Baek, Aaron Park, Young-Jin Ahna and Jaebum Choo: ``Baseline correction using asymmetrically reweighted penalized least squares smoothing'', Analyst, 2015,140, 250-257 <<https://pubs.rsc.org/en/content/articlelanding/2015/an/c4an01061b>>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** Rcpp (>= 1.0.13), limSolve

**LinkingTo** Rcpp, RcppArmadillo

**RoxygenNote** 7.3.2

**Depends** R (>= 2.10)

**Suggests** knitr,  
rmarkdown

**VignetteBuilder** knitr

## R topics documented:

Abelsonite . . . . .	2
baseline_estimation . . . . .	2
new_arPLSresult . . . . .	4
plot.arPLSresult . . . . .	4
RcppArmadillo-Functions . . . . .	5
rcpparma_armaInv . . . . .	6
summary.arPLSresult . . . . .	6
<b>Index</b>	<b>7</b>

---

Abelsonite	<i>Raw Raman spectrum for Abelsonite A data frame containing 3315 rows and 2 variables (wavenumber and measurement)</i>
------------	---

---

### Description

Raw Raman spectrum for Abelsonite A data frame containing 3315 rows and 2 variables (wavenumber and measurement)

### Usage

Abelsonite

### Format

An object of class `data.frame` with 3315 rows and 2 columns.

### Author(s)

Bob Downs <rdowns@u.arizona.edu>

### Source

[https://rruff.info/repository/sample\\_child\\_record\\_raman\\_full/by\\_minerals/Abelsonite\\_R070007\\_\\_Broad\\_Scan\\_\\_532\\_\\_0\\_\\_unoriented\\_\\_Raman\\_Data\\_RAW\\_\\_13756.txt](https://rruff.info/repository/sample_child_record_raman_full/by_minerals/Abelsonite_R070007__Broad_Scan__532__0__unoriented__Raman_Data_RAW__13756.txt)

### References

Lafuente B, Downs R T, Yang H, Stone N (2015) The power of databases: the RRUFF project. In: Highlights in Mineralogical Crystallography, T Armbruster and R M Danisi, eds. Berlin, Germany, W. De Gruyter, pp 1-30

---

baseline_estimation	<i>asymmetrically reweighted penalized least squares</i>
---------------------	--

---

### Description

Baseline estimation using asymmetrically reweighted penalized least squares smoothing (Baek et al. 2015).

### Usage

```
baseline_estimation(
  y,
  lambda = 1e+06,
  ratio = 1e-06,
  max_iter = 50,
  verbose = FALSE,
  algo = "banded"
)
```

**Arguments**

<code>y</code>	Numeric vector representing the spectrum.
<code>lambda</code>	Smoothing parameter. The smaller the more curvature (wiggleness). (default: 1e6).
<code>ratio</code>	Stopping criterion based on changes in weight vector per iteration (default: 1e-6).
<code>max_iter</code>	Maximum number of iterations as fall back criterion if no conversion happens (default: 50).
<code>verbose</code>	Boolean to print intermediary outputs (default: FALSE).
<code>algo</code>	String to choose solver between Armadillo CPP <code>armaInv</code> ("cpp") and native solver function "native" and <code>limSolve::Solve.banded</code> solver ("banded") (default: "banded").

**Details**

The algorithm iteratively estimates a spectral baseline curve by updating a weight vector by means of a generalized logistic function that focuses the estimation efforts on regions where the baseline and the signal are close to each other

**Value**

object of class `arPLSresult`:

- `rawinput`: The original spectrum fed into the algorithm.
- `lambda`: The lambda parameter fed into the algorithm.
- `ratio`: The ratio stopping parameter fed into the algorithm.
- `max_iter`: The maximum iteration stopping parameter fed into the algorithm.
- `baseline`: The fitted spectral baseline.
- `last_iter`: The number of iterations the algorithm did before stopping.
- `last_ratio`: The last value of the ratio stopping criterium before stopping.

**Author(s)**

Corvin Idler

**References**

Baek, S.-J., Park, A., Ahn, Y.-J., and Choo, J. (2015). Baseline correction using asymmetrically reweighted penalized least squares smoothing. *Analyst*, 140:250–257.

**Examples**

```
{
y <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
baseline <- baseline_estimation(y)
}
```

---

new_arPLSresult	<i>Constructor for arPLSresult object</i>
-----------------	---

---

### Description

This is an constructor for the S3 object arPLSresult.

### Usage

```
new_arPLSresult(
  rawinput = numeric(),
  lambda = 1e+06,
  ratio = 1e-06,
  max_iter = 50,
  baseline = numeric(),
  last_iter = integer(),
  last_ratio = double()
)
```

### Arguments

rawinput	The original spectrum fed into the algorithm.
lambda	The lambda parameter fed into the algorithm.
ratio	The ratio stopping parameter fed into the algorithm.
max_iter	The maximum iteration stopping parameter fed into the algorithm.
baseline	The fitted spectral baseline.
last_iter	The number of iterations the algorithm did before stopping.
last_ratio	The last value of the ratio stopping criterium before stopping.

---

plot.arPLSresult	<i>Take an object of class arPLSresult and plot some results</i>
------------------	--

---

### Description

This is an S3 generic. To plot an input spectrum and an estimated baseline spectrum.

### Usage

```
## S3 method for class 'arPLSresult'
plot(x, ...)
```

### Arguments

x	A result object of class arPLSresult (mainly a list).
...	placeholder for arbitrary additional parameters (to stay in line with other generic plot functions)

---

**RcppArmadillo-Functions***Set of functions in example RcppArmadillo package*

---

**Description**

These four functions are created when `RcppArmadillo.package.skeleton()` is invoked to create a skeleton packages.

**Usage**

```
rcpparma_hello_world()  
rcpparma_outerproduct(x)  
rcpparma_innerproduct(x)
```

**Arguments**

x	a numeric vector
---	------------------

**Details**

These are example functions which should be largely self-explanatory. Their main benefit is to demonstrate how to write a function using the Armadillo C++ classes, and to have to such a function accessible from R.

**Value**

`rcpparma_hello_world()` does not return a value, but displays a message to the console.

`rcpparma_outerproduct()` returns a numeric matrix computed as the outer (vector) product of x.

`rcpparma_innerproduct()` returns a double computer as the inner (vector) product of x.

**Author(s)**

Dirk Eddelbuettel

**References**

See the documentation for Armadillo, and RcppArmadillo, for more details.

**Examples**

```
x <- sqrt(1:4)  
rcpparma_innerproduct(x)  
rcpparma_outerproduct(x)
```

---

rcpparma_armaInv	<i>Title Armadillo package matrix inversion function Description Takes a matrix and inverts it.</i>
------------------	---

---

**Description**

Title Armadillo package matrix inversion function Description Takes a matrix and inverts it.

**Usage**

```
rcpparma_armaInv(x)
```

**Arguments**

x	matrix to be inverted
---	-----------------------

**Value**

Inverted matrix

---

summary.arPLSresult	<i>Take an object of class arPLSresult and summarize (print) some facts about it</i>
---------------------	--

---

**Description**

This is an S3 generic. To summarize (print) some facts about the arPLS baseline estimation that led to it.

**Usage**

```
## S3 method for class 'arPLSresult'
summary(object, ...)
```

**Arguments**

object	A result object of class arPLSresult (mainly a list).
...	placeholder for arbitrary additional parameters (to stay in line with other generic summary functions)

# Index

## \* datasets

Abelsonite, [2](#)

Abelsonite, [2](#)

baseline\_estimation, [2](#)

new\_arPLSresult, [4](#)

plot.arPLSresult, [4](#)

rcpparma\_armaInv, [6](#)

rcpparma\_hello\_world  
(RcppArmadillo-Functions), [5](#)

rcpparma\_innerproduct  
(RcppArmadillo-Functions), [5](#)

rcpparma\_outerproduct  
(RcppArmadillo-Functions), [5](#)

RcppArmadillo-Functions, [5](#)

summary.arPLSresult, [6](#)