

# Package ‘ISEtools’

May 7, 2026

**Type** Package

**Title** Ion Selective Electrodes Analysis Methods

**Version** 3.2.1

**Date** 2025-12-22

**Author** Peter Dillingham [aut, cre],  
Christina McGraw [ctb],  
Aleksandar Radu [ctb],  
Basim Alsaedi [ctb]

**Maintainer** Peter Dillingham <peter.dillingham@otago.ac.nz>

**Description** Characterisation and calibration of single or multiple Ion Selective Electrodes (ISEs); activity estimation of experimental samples. Implements methods described in:

Dillingham, P.W., Radu, T., Diamond, D., Radu, A. and McGraw, C.M. (2012) <[doi:10.1002/elan.201100510](https://doi.org/10.1002/elan.201100510)>,

Dillingham, P.W., Alsaedi, B.S.O. and McGraw, C.M. (2017) <[doi:10.1109/ICSENS.2017.8233898](https://doi.org/10.1109/ICSENS.2017.8233898)>,

Dillingham, P.W., Alsaedi, B.S.O., Radu, A., and McGraw, C.M. (2019) <[doi:10.3390/s19204544](https://doi.org/10.3390/s19204544)>, and

Dillingham, P.W., Alsaedi, B.S.O., Granados-Focil, S., Radu, A., and McGraw, C.M. (2020) <[doi:10.1021/acssensors.9b02133](https://doi.org/10.1021/acssensors.9b02133)>.

**Depends** R (>= 4.1.0)

**Imports** coda

**Suggests** R2OpenBUGS, rjags, boot, R.rsp

**VignetteBuilder** R.rsp

**SystemRequirements** OpenBUGS (>=3.0) or JAGS (>=4.3.1)

**License** GPL-2

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2026-01-09 09:50:02 UTC

## Contents

ISEtools-package	2
analyseISE	3
carbonate	6
describeISE	6
LeadStdAdd	9
loadISEdata	10
plot.analyseISE	12
plot.ISEdata	14
plot.ISEdescription	15
print.analyseISE	15
print.ISEdata	16
print.ISEdescription	17
summary.analyseISE	17
summary.ISEdata	18
summary.ISEdescription	19

<b>Index</b>	<b>20</b>
--------------	-----------

---

ISEtools-package	<i>Ion Selective Electrodes Analysis Methods</i>
------------------	--

---

## Description

Bayesian calibration for single or multiple ISEs using R and OpenBUGS (or JAGS). Estimation of analyte activities using single ISEs or ISE arrays.

## Details

Package:	ISEtools
Type:	Package
Version:	3.2.1
Depends: R (>4.1.0)	
Date:	2025-12-22
License:	GPL-2
SystemRequirements: OpenBUGS (>3.0) or JAGS (>=4.3.1)	

The primary functions are loadISEdata (which loads calibration and experimental data from tab-delimited text files), describeISE (uses Bayesian calibration to estimate ISE parameters from calibration data), and analyseISE (combines calibration data with experimental data in basic or standard addition format to estimate analyte concentrations).

## Author(s)

Peter Dillingham [aut, cre], Christina McGraw [ctb], Aleksandar Radu [ctb], Basim Alsaedi [ctb]  
 Maintainer: Peter Dillingham <peter.dillingham@otago.ac.nz>

## References

- Dillingham, P.W., Radu, T., Diamond, D., Radu, A. and McGraw, C.M. (2012). Bayesian Methods for Ion Selective Electrodes. *Electroanalysis*, **24**, 316-324. <doi:10.1002/elan.201100510>
- Dillingham, P.W., Alsaedi, B.S.O. and McGraw, C.M. (2017). Characterising uncertainty in instrumental limits of detection when sensor response is non-linear. *2017 IEEE SENSORS*, Glasgow, United Kingdom, pp. 1-3. <doi:10.1109/ICSENS.2017.8233898>
- Dillingham, P.W., Alsaedi, B.S.O., Radu, A., and McGraw, C.M. (2019). Semi-automated data analysis for ion-selective electrodes and arrays using the R package ISEtools. *Sensors* **19**(20), 4544. <doi:10.3390/s19204544>
- Dillingham, P.W., Alsaedi, B.S.O., Granados-Focil, S., Radu, A., and McGraw, C.M. (2020). Establishing meaningful Limits of Detection for ion-selective electrodes and other nonlinear sensors *ACS Sensors*, **5**, 250-257. <doi:10.1021/acssensors.9b02133>

## Examples

```
data(LeadStdAdd)
print(LeadStdAdd)
summary(LeadStdAdd)
plot(LeadStdAdd)
```

---

analyseISE	<i>Ion selective electrode characterisation and estimation of sample concentrations</i>
------------	---

---

## Description

Use Bayesian calibration to estimate parameters for  $y = a + b \log(x + c) + \text{error}$ , where error follows a normal distribution with mean 0 and standard deviation  $\sigma$ . The limit of detection (false positive/negative method or  $S/N=3$  method) is also estimated. These values are then used to the estimate sample concentrations.

## Usage

```
analyseISE(data, model.path=NA, model.name=NA, Z=NA, temperature = 21,
burnin=25000, iters = 50000, chains=4, thin = 1,
a.init= NA, b.init=NA, cstar.init=NA, logc.limits = c(-8.9, -1.9),
sigma.upper = 5, diagnostic.print=FALSE, offset = 1,
alpha = 0.05, beta = 0.05, SN = NA, program="OpenBUGS")
```

## Arguments

data	Calibration and experimental data (of class 'ISEdata'; see loadISEdata)
model.path	The directory where the BUGS model is located (defaults to 'models' sub-directory under the location of ISEtools package, e.g. '../ISEtools/models')
model.name	The name of the BUGS model (e.g. 'Single_ISE_model.txt') (defaults are located in ISEtools package)

Z	Ionic valence (e.g. for lead, Z = 2)
temperature	temperature in degrees C
burnin	Initial number of Monte Carlo simulations to discard.
iters	Total number of iterations.
chains	Number of parallel MCMC chains
thin	Thinning rate, equal to 1/Proportion of simulations saved (e.g. thin = 10 records every tenth iteration).
a.init	Initial value for parameter a
b.init	Initial value for parameter b
cstar.init	Initial value for parameter cstar ( $c = cstar^{10}$ )
logc.limits	Upper and lower limits for log c initial values
sigma.upper	Upper limit for initial value of sigma
diagnostic.print	logical flag indicating whether a diagnostic printout is desired (default is F)
offset	The initial value for the slope is based on the last data point as sorted by concentration (i.e. the Nth point) and the (N - offset) data point. The default is offset = 1, corresponding to the last and second to last data points.
alpha	False positive rate used for detection threshold (not output) to calculate LOD(alpha, beta) (only returned if SN = NA)
beta	False negative rate used to calculate LOD(alpha, beta) (only returned if SN = NA)
SN	Desired signal-to-noise ratio for LOD(S/N) calculations (default is to calculate the S/N equivalent based on alpha, beta)
program	Choice of "OpenBUGS" (default and recommended for Windows or Linux) or "jags" (for macOS, see manual for warnings).

### Value

analyseISE returns a list of class 'analyseISE'. Individual components include:

SampleID	Sample identification number
log10x.exp	Estimated concentration (log scale, mol/l)
ahat	Estimated value for a (from the median of the posterior distribution)
bhat	Estimated value for b (from the median of the posterior distribution)
chat	Estimated value for c (from the median of the posterior distribution)
cstarhat	Estimated value for cstar (from the median of the posterior distribution)
sigmahat	Estimated value for sigma (from the median of the posterior distribution)
LOD.info	List describing LOD method (alpha, beta or S/N) and corresponding values (alpha, beta, SN)
LOD.hat	Estimated value for the limit of detection (from the median of the posterior distribution)

<code>&lt;parametername&gt;.lcl</code>	Lower limit for the above parameters (e.g. <code>ahat.lcl</code> , <code>bhat.lcl</code> , ...) (from the 2.5th percentile of the posterior distribution)
<code>&lt;parametername&gt;.ucl</code>	Upper limit for the above parameters (from the 97.5th percentile of the posterior distribution)
<code>LOD.Q1</code>	25th percentile estimated value of the limit of detection
<code>LOD.Q3</code>	75th percentile estimated value of the limit of detection

### Author(s)

Peter Dillingham, <peter.dillingham@otago.ac.nz>

### References

Dillingham, P.W., Radu, T., Diamond, D., Radu, A. and McGraw, C.M. (2012). Bayesian Methods for Ion Selective Electrodes. *Electroanalysis*, **24**, 316-324. <doi:10.1002/elan.201100510>

Dillingham, P.W., Alsaedi, B.S.O. and McGraw, C.M. (2017). Characterising uncertainty in instrumental limits of detection when sensor response is non-linear. *2017 IEEE SENSORS*, Glasgow, United Kingdom, pp. 1-3. <doi:10.1109/ICSENS.2017.8233898>

Dillingham, P.W., Alsaedi, B.S.O., Radu, A., and McGraw, C.M. (2019). Semi-automated data analysis for ion-selective electrodes and arrays using the R package ISEtools. *Sensors* **19**(20), 4544. <doi: 10.3390/s19204544>

Dillingham, P.W., Alsaedi, B.S.O., Granados-Focil, S., Radu, A., and McGraw, C.M. (2020). Establishing meaningful Limits of Detection for ion-selective electrodes and other nonlinear sensors *ACS Sensors*, **5**, 250-257. <doi:10.1021/acssensors.9b02133>

### Examples

```
# Fast-running example with only 100 MCMC iterations for testing:
data(LeadStdAdd)
if (requireNamespace("rjags", quietly = TRUE)) {
  example2test = analyseISE(LeadStdAdd, Z = 2, temperature = 21,
    burnin=100, iters=200, chains=1, a.init=c(176, 146, -112),
    b.init=c(29, 30, 31), cstar.init=c(0.26, 0.27, 0.22), program="jags")
  print(example2test)
  summary(example2test)
  plot(example2test, ylim = c(-7, -3), xlab = "ID (Sample)",
    ylab = expression(paste(log[10], " ", Pb^{paste("2","+",sep="" )})) )
} else {
  message("Package 'rjags' is not installed, skipping JAGS example.")
}
```

---

 carbonate

*ISE measurements of carbonate in seawater*


---

**Description**

A data set containing emf responses for 8 ISEs measuring carbonate in seawater

**Usage**

```
data(carbonate)
```

**Format**

Load example carbonate data as an object of type ISEdata (see function loadISEdata)

**References**

Dillingham, P.W., Alsaedi, B.S.O. and McGraw, C.M. (2017). Characterising uncertainty in instrumental limits of detection when sensor response is non-linear. *2017 IEEE SENSORS*, Glasgow, United Kingdom, pp. 1-3. <doi:10.1109/ICSENS.2017.8233898>

**Examples**

```
data(carbonate)
print(carbonate)
plot(carbonate)
```

---

 describeISE

*Ion selective electrode characterisation*


---

**Description**

Use Bayesian calibration to estimate parameters for  $y = a + b \log(x + c) + \text{error}$ , where error follows a normal distribution with mean 0 and standard deviation  $\sigma$ . The limit of detection is also estimated.

**Usage**

```
describeISE(data, model.path=NA, model.name = NA, Z=NA, temperature = 21,
burnin=25000, iters = 50000, chains=4, thin = 1,
a.init= NA, b.init=NA, cstar.init=NA,
logc.limits = c(-8.9, -1.9), sigma.upper = 5, diagnostic.print=FALSE, offset = 1,
alpha = 0.05, beta = 0.05, SN = NA,
keep.coda=TRUE, coda.n=1000, program="OpenBUGS")
```

**Arguments**

<code>data</code>	Calibration data (of class 'ISEdata'; see loadISEdata)
<code>model.path</code>	The directory where the BUGS model is located (defaults to 'models' sub-directory under the location of ISEtools package, e.g. '.../ISEtools/models')
<code>model.name</code>	The name of the BUGS model (e.g. 'Single_ISE_model.txt') (defaults are located in ISEtools package)
<code>Z</code>	Ionic valence (e.g. for lead, $Z = 2$ )
<code>temperature</code>	temperature in degrees C
<code>burnin</code>	Initial number of Monte Carlo simulations to discard.
<code>iters</code>	Total number of iterations.
<code>chains</code>	Number of parallel MCMC chains
<code>thin</code>	Thinning rate, equal to 1/Proportion of simulations saved (e.g. <code>thin = 10</code> records every tenth iteration).
<code>a.init</code>	Initial value for parameter a
<code>b.init</code>	Initial value for parameter b
<code>cstar.init</code>	Initial value for parameter cstar ( $c = cstar^{10}$ )
<code>logc.limits</code>	Upper and lower limits for log c initial values
<code>sigma.upper</code>	Upper limit for initial value of sigma
<code>diagnostic.print</code>	logical flag indicating whether a diagnostic printout is desired (default is FALSE)
<code>offset</code>	The initial value for the slope is based on the last data point as sorted by concentration (i.e. the Nth point) and the (N - offset) data point. The default is <code>offset = 1</code> , corresponding to the last and second to last data points.
<code>alpha</code>	False positive rate used for detection threshold (not output) to calculate LOD(alpha, beta) (only returned if SN = NA)
<code>beta</code>	False negative rate used to calculate LOD(alpha, beta) (only returned if SN = NA)
<code>SN</code>	Desired signal-to-noise ratio for LOD(S/N) calculations (default is to calculate the S/N equivalent based on alpha, beta)
<code>keep.coda</code>	Logical flag indicating whether the MCMC simulations should be returned ( <code>keep.coda = TRUE</code> ) or not ( <code>keep.coda = FALSE</code> )
<code>coda.n</code>	Indicates how many simulations to return (sampled with replacement). If <code>coda.n &gt;=</code> the total, all are returned.
<code>program</code>	Choice of "OpenBUGS" (default and recommended for Windows or Linux) or "jags" (for macOS, see manual for warnings).

**Value**

describeISE returns a list of class 'ISEdescription'. Individual components are:

<code>ahat</code>	Estimated value for a (from the median of the posterior distribution)
-------------------	---

bhat	Estimated value for b (from the median of the posterior distribution)
chat	Estimated value for c (from the median of the posterior distribution)
cstarhat	Estimated value for cstar (c to the 0.1 power) (from the median of the posterior distribution)
sigmahat	Estimated value for cstar (from the median of the posterior distribution)
LOD.info	List describing LOD method (alpha, beta or S/N) and corresponding values (alpha, beta, SN)
LOD.hat	Estimated value for the limit of detection (from the median of the posterior distribution)
<parametername>.lcl	Lower limit for the above parameters (e.g. ahat.lcl, bhat.lcl, ...) (from the 2.5th percentile of the posterior distribution)
<parametername>.ucl	Upper limit for the above parameters (from the 95.5th percentile of the posterior distribution)
LOD.Q1	25th percentile estimated value of the limit of detection
LOD.Q3	75th percentile estimated value of the limit of detection

If keep.coda = TRUE, then these additional items are returned:

ahat.coda	Random sample (without replacement) of length coda.n from the Markov Chain Monte Carlo simulations for a
bhat.coda	Random sample (without replacement) of length coda.n from the Markov Chain Monte Carlo simulations for b
chat.coda	Random sample (without replacement) of length coda.n from the Markov Chain Monte Carlo simulations for c
sigmahat.coda	Random sample (without replacement) of length coda.n from the Markov Chain Monte Carlo simulations for sigma
cstarhat.coda	Random sample (without replacement) of length coda.n from the Markov Chain Monte Carlo simulations for cstar
LOD.coda	Random sample (without replacement) of length coda.n from the Markov Chain Monte Carlo simulations for LOD

### Author(s)

Peter Dillingham, <peter.dillingham@otago.ac.nz>

### References

- Dillingham, P.W., Radu, T., Diamond, D., Radu, A. and McGraw, C.M. (2012). Bayesian Methods for Ion Selective Electrodes. *Electroanalysis*, **24**, 316-324.
- Dillingham, P.W., Alsaedi, B.S.O. and McGraw, C.M. (2017). Characterising uncertainty in instrumental limits of detection when sensor response is non-linear. *2017 IEEE SENSORS*, Glasgow, United Kingdom, pp. 1-3. <doi:10.1109/ICSENS.2017.8233898>

Dillingham, P.W., Alsaedi, B.S.O., Radu, A., and McGraw, C.M. (2019). Semi-automated data analysis for ion-selective electrodes and arrays using the R package ISEtools. *Sensors* **19**(20), 4544. <doi: 10.3390/s19204544>

Dillingham, P.W., Alsaedi, B.S.O., Granados-Focil, S., Radu, A., and McGraw, C.M. (2020). Establishing meaningful Limits of Detection for ion-selective electrodes and other nonlinear sensors *ACS Sensors*, **5**, 250-257. <doi:10.1021/acssensors.9b02133>

## Examples

```
# Fast-running example with only 100 MCMC iterations for testing:
data(carbonate)
if (requireNamespace("rjags", quietly = TRUE)) {
  example3test = describeISE(carbonate, Z = -2, SN = 3.6,
    burnin=100, iters=200, chains=1,
    a.init= c(-50,180,140,65,100,170,100,130),
    b.init=rep(-20,8), cstar.init=rep(0.2, 8), program="jags")
  print(example3test)
  summary(example3test)
  plot(example3test)
} else {
  message("Package 'rjags' is not installed, skipping JAGS example.")
}
```

---

LeadStdAdd

*ISE measurements of lead in soil*

---

## Description

A data set containing emf responses for 3 ISEs measuring lead in soil at Silvermines, Ireland. Calibration data and experimental data for 17 samples (in standard addition format) are included.

## Usage

```
data(LeadStdAdd)
```

## Format

Load example lead data as an object of type ISEdata (see function loadISEdata)

## References

Dillingham, P.W., Radu, T., Diamond, D., Radu, A. and McGraw, C.M. (2012). Bayesian Methods for Ion Selective Electrodes. *Electroanalysis*, **24**, 316-324. <doi:10.1002/elan.201100510>

Dillingham, P.W., Alsaedi, B.S.O., Radu, A., and McGraw, C.M. (2019). Semi-automated data analysis for ion-selective electrodes and arrays using the R package ISEtools. *Sensors* **19**(20), 4544. <doi: 10.3390/s19204544>

Dillingham, P.W., Alsaedi, B.S.O., Granados-Focil, S., Radu, A., and McGraw, C.M. (2020). Establishing meaningful Limits of Detection for ion-selective electrodes and other nonlinear sensors. *ACS Sensors*, **5**, 250-257. <doi:10.1021/acssensors.9b02133>

**Examples**

```
data(LeadStdAdd)
print(LeadStdAdd)
summary(LeadStdAdd)
plot(LeadStdAdd)
```

---

loadISEdata

*Load ISE calibration and experimental data.*


---

**Description**

Loads tab-delimited calibration and (if it exists) experimental sample data.

**Usage**

```
loadISEdata(filename.calibration, filename.experimental = NA)
```

**Arguments**

filename.calibration

The name and location of the tab-delimited calibration file

It should have the following structure:

First line (header row): ISEID log10x emf

Remaining lines (data): ISEID is an identifier for the ISE. The ISEID variables should be integers, with the lowest value equal to 1, and no gaps. That is, if there are four ISEs, they must be labeled 1, 2, 3, and 4. log10x is the log10 concentration (mol/l) of the calibration samples. The emf readings (in mV) follow.

filename.experimental

The experimental file (if there is one, otherwise keep the default filename.experimental=NA) should have one of the following structures:

basic model: The header row will include ISEID, SampleID, and emf. ISEID is defined the same way as in the calibration file. SampleID is an integer indicating which sample is being measured, and must follow the same numbering rules as ISEID. Finally, emf is the mV reading of the experimental samples for each ISE.  
or

standard addition: When using the standard addition model, the experimental file will contain ISEID and SampleID as before. Two emf values are recorded: emf1 is the mV reading of the sample, and emf2 is the mV reading of the sample plus the addition. Additionally, V.s is the volume of the sample, V.add is the volume of the addition, and conc.add is the concentration (mol/l) of the addition. The units of V.s and V.add do not matter as long as they are the same.

**Details**

Internally calls 'ISEdata.calibration' if there is no experimental data.

**Value**

loadISEdata returns the following values in a list of class ISEdata:

Calibration variables:

N	Total number of calibration measurements (e.g. for 5 calibration points measured with 3 ISEs, N = 15)
R	Number of ISEs
ISEID	Identifier for the ISE
log10x	log concentration (mol/l) of calibration data
emf	emf (mV) for calibration data

Experimental variables:

M	Number of experimental samples
M.obs	Total number of experimental measurements. E.g. for 4 samples each measured by 3 ISEs, M.obs = 12. Only returned if R > 1
ISEID.exp	Identifier for the ISE for the experimental data (returned if R > 1)
x.exp	Identifier for the experimental (returned if R > 1)

Basic format only:

emf.exp	emf (mV) for experimental data
---------	--------------------------------

Standard addition format only:

delta.emf	difference between emf1 and emf2 (mV) for experimental data
V.s	Sample volume (any units allowed but must be consistent)
V.add	Volume added to the sample
conc.add	Concentration added.

Summary variables of calibration and experimental data:

calibration.only	Indicates whether there was only calibration data (TRUE) or calibration and experimental data (FALSE)
stdadd	Indicates whether standard addition was used. Returns NA (calibration data only), FALSE (basic experimental data), or TRUE (standard addition experimental data)
data.calib	The loaded calibration data frame
data.exp	The loaded experimental data frame

**Author(s)**

Peter Dillingham <peter.dillingham@otago.ac.nz>

**Examples**

```
###
# Loading the example tab-delimited text files for the lead data
###

# 1. Find pathnames for the lead example txt files:
path.calib = paste(path.package('ISEtools'), "/extdata",
"/Lead_calibration.txt", sep="")
path.basic = paste(path.package('ISEtools'), "/extdata",
"/Lead_experimentalBasic.txt", sep="")
path.sa = paste(path.package('ISEtools'), "/extdata",
"/Lead_experimentalSA.txt", sep="")
# Load the calibration data
lead.example1 = loadISEdata(filename.calibration = path.calib)
print(lead.example1)

# ... and with experimental data, Basic format
lead.example2 = loadISEdata(filename.calibration = path.calib,
filename.experimental = path.basic)
print(lead.example2)

# ... and with experimental data, Standard Addition format
lead.example3 = loadISEdata(filename.calibration = path.calib,
filename.experimental = path.sa)
print(lead.example3)
```

---

plot.analyseISE

*Plot function for ion selective electrode characterisation and estimation of sample concentrations*

---

**Description**

Plots sample concentration estimates derived from Bayesian calibration. E.g. analyseISE uses Bayesian calibration to estimate parameters for  $y = a + b \log(x + c) + \text{error}$ , where error follows a normal distribution with mean 0 and standard deviation sigma. These values are combined with experimental data to estimate sample concentrations.

**Usage**

```
## S3 method for class 'analyseISE'
plot(x, xlab = "Sample ID",
      ylab = expression(paste(log[10], " { ", italic(x), " }")), xlim = NA,
      ylim = c(-15, 0), x.ticks = NA, y.ticks = NA, x.ticks.label = TRUE,
      y.ticks.label = TRUE, y.las = 2, col = 1, x.shift = 0, xaxs = "r",
      yaxs = "r", add.box = TRUE, ...)
```

**Arguments**

x	Calibration and experimental sample results (of class 'analyseISE'; see analyseISE)
xlab	Label for the x-axis
ylab	Label for the y-axis
xlim	Limits for the x-axis. Automatically calculated if xlim = NA.
ylim	Limits for the y-axis.
x.ticks	Location of tickmarks for the x-axis. Automatically calculated if x.ticks = NA.
y.ticks	Location of tickmarks for the y-axis. Automatically calculated if y.ticks = NA.
x.ticks.label	Labels associated with x-axis tickmarks for the x-axis. Automatically calculated labels (TRUE), no labels (FALSE), or a column of text specifying custom labels (e.g. x.ticks.label = c("A", "B", "C") or similar, of the same length as x.ticks).
y.ticks.label	Labels associated with y-axis tickmarks for the y-axis. See x.ticks.label for details.
y.las	Indicates whether y-axis labels be perpendicular to the y-axis (2) or parallel to it (0).
col	Colour for the field of the plot.
x.shift	Shifts the plots to the left (- values) or right (+ values); useful for overlaying figures.
xaxs	The style of x-axis interval. See par for further details, but "r" adds 4 percent padding, "i" has no padding.
yaxs	The style of y-axis interval. See yaxs above.
add.box	Indicates whether a box should be drawn around the plot (TRUE) or not (FALSE).
...	Other arguments to be passed through to plotting functions.

**Value**

No return value, creates plot.

**Author(s)**

Peter Dillingham, <peter.dillingham@otago.ac.nz>

**See Also**

[analyseISE](#)

---

`plot.ISEdata`*Basic plot of ion selective electrode calibration data*

---

**Description**

Plots raw ISE calibration data; data should follow a hockey stick pattern coinciding with the equation  $y = a + b \log(x + c) + \text{error}$ , where error follows a normal distribution with mean 0 and standard deviation  $\sigma$ .

**Usage**

```
## S3 method for class 'ISEdata'
plot(x, xlab = expression(paste(log[10], " { ", italic(x),
  " }")), ylab = "emf", pch = 20, ...)
```

**Arguments**

<code>x</code>	ISE calibration data
<code>xlab</code>	Label for the x-axis
<code>ylab</code>	Label for the y-axis
<code>pch</code>	Plotting symbol for data
<code>...</code>	Other arguments to be passed through to plotting functions.

**Value**

No return value, creates plot.

**Author(s)**

Peter Dillingham, <peter.dillingham@otago.ac.nz>

**See Also**

[loadISEdata](#)

**Examples**

```
data(LeadStdAdd)
plot(LeadStdAdd)
```

---

plot.ISEdescription     *Plot ISE parameter values*

---

**Description**

Plots histograms of ISE parameter values a, b, c, sigma, and LOD (alpha, beta or S/N) for the equation  $y = a + b \log(x + c) + \text{error}$ , where error follows a normal distribution with mean 0 and standard deviation sigma.

**Usage**

```
## S3 method for class 'ISEdescription'  
plot(x, breaks = 20, ...)
```

**Arguments**

x	ISE description (e.g. object of class ISEdescription)
breaks	Approximate number of bins for histograms, defaults to 20
...	Other arguments to be passed through to plotting (histogram) functions

**Value**

No return value, creates plot.

**Author(s)**

Peter Dillingham, <peter.dillingham@otago.ac.nz>

**See Also**

[describeISE](#)

---

print.analyseISE     *Prints tables of ISE parameters and estimated sample concentrations.*

---

**Description**

Prints tables of ISE parameters and estimated sample concentrations.

**Usage**

```
## S3 method for class 'analyseISE'  
print(x, ...)
```

**Arguments**

x                    ISE analysis results (e.g. object of class analyseISE)  
...                  Other objects passed through.

**Value**

No return value, prints results from analyseISE.

**Author(s)**

Peter Dillingham, <peter.dillingham@otago.ac.nz>

**See Also**

[analyseISE](#)

---

print.ISEdata	<i>Prints ISE data</i>
---------------	------------------------

---

**Description**

Prints tables of calibration data and experimental data (if present).

**Usage**

```
## S3 method for class 'ISEdata'  
print(x, ...)
```

**Arguments**

x                    ISE data (e.g. object of class ISEdata)  
...                  Other objects passed through.

**Value**

No return value, prints ISE data.

**Author(s)**

Peter Dillingham, <peter.dillingham@otago.ac.nz>

**See Also**

[loadISEdata](#)

**Examples**

```
data(LeadStdAdd)  
print(LeadStdAdd)
```

---

print.ISEdescription    *Prints tables of ISE parameters.*

---

**Description**

Prints tables of ISE parameters for one or multiple ISEs.

**Usage**

```
## S3 method for class 'ISEdescription'  
print(x, ...)
```

**Arguments**

x	ISE analysis results (e.g. object of class analyseISE)
...	Other objects passed through.

**Value**

No return value, prints results from describeISE.

**Author(s)**

Peter Dillingham, <peter.dillingham@otago.ac.nz>

**See Also**

[describeISE](#)

---

summary.analyseISE    *Summary of estimates for ISE parameter values and experimental sample concentrations.*

---

**Description**

summary.analyseISE takes an object of class analyseISE and produces summary tables.

**Usage**

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

**Arguments**

object	Data set of class ISEdata
...	Other objects passed through.

**Value**

tables: Two tables (table1 and table2) are returned as a list.

table1	A table of ISE parameter values (see summary.describeISE for details)
table2	A table of estimated analyte concentrations for experimental samples

**Author(s)**

Peter Dillingham, <peter.dillingham@otago.ac.nz>

**See Also**

[analyseISE](#) [summary.ISEdescription](#)

---

summary.ISEdata	<i>Summarises ISE data</i>
-----------------	----------------------------

---

**Description**

summary.ISE takes an object of class ISEdata (e.g. see loadISEdata) and produces metadata for it.

**Usage**

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

**Arguments**

object	Data set of class ISEdata
...	Other objects passed through.

**Value**

metadata: Metadata for the ISEs, a list with N, R, calibration.only, M, and stdadd

N	Total number of calibration observations
R	Number of ISEs
calibration.only	Indicates calibration only data (T), or calibration and experimental data (F)
M	Number of experimental samples (NA if no experimental data were loaded)
stdadd	Indicates whether standard addition used for experimental samples (T) or the basic model was used (F), or no experimental data (NA)

**Author(s)**

Peter Dillingham, <peter.dillingham@otago.ac.nz>

**See Also**[loadISEdata](#)**Examples**

```
data(LeadStdAdd)
summary(LeadStdAdd)
```

---

```
summary.ISEdescription
Summarise ISE parameters
```

---

**Description**

summary.ISEdescription takes an object of class ISEdescription and prints a table of parameter values for  $y = a + b \log(x + c) + \text{error}$ , with the errors following a Normal distribution with mean 0 and standard deviation sigma. Also calculates LOD using the conditional analytic method (alpha, beta, or S/N).

**Usage**

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

**Arguments**

object	object of class ISEdescription
...	Other objects passed through.

**Value**

table1: A matrix with parameter values for each ISE

**Author(s)**

Peter Dillingham, <peter.dillingham@otago.ac.nz>

**See Also**[describeISE](#)

# Index

## \* datasets

carbonate, [6](#)

LeadStdAdd, [9](#)

## \* package

ISEtools-package, [2](#)

analyseISE, [3](#), [13](#), [16](#), [18](#)

carbonate, [6](#)

describeISE, [6](#), [15](#), [17](#), [19](#)

ISEtools (ISEtools-package), [2](#)

ISEtools-package, [2](#)

LeadStdAdd, [9](#)

loadISEdata, [10](#), [14](#), [16](#), [19](#)

plot.analyseISE, [12](#)

plot.ISEdata, [14](#)

plot.ISEdescription, [15](#)

print.analyseISE, [15](#)

print.ISEdata, [16](#)

print.ISEdescription, [17](#)

summary.analyseISE, [17](#)

summary.ISEdata, [18](#)

summary.ISEdescription, [18](#), [19](#)