

# Using R for Bayesian calibration of ISE data

Bayesian calibration of ion selective electrode (ISE) data uses the free, open-source program, **OpenBUGS**, to perform analyses. For some users, however, it is more convenient to interface to OpenBUGS through a different software package that they are more familiar with, or that has more general use, such as **R**. This tutorial shows how to use supplied R functions that can be used in combination with the R2WinBUGS library.

## 1. Installation: R ([www.r-project.org](http://www.r-project.org)) and OpenBUGS

(<http://mathstat.helsinki.fi/openbugs/>) must be installed on your computer. The R packages, **R2WinBUGS**, **coda**, and **lattice** must also be installed.

### 1a. Installing R

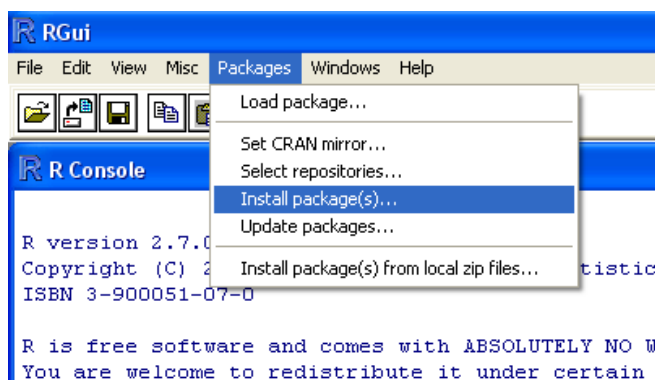
- In order to install R, a **mirror** must be selected. From the R homepage, click on [CRAN](http://cran.r-project.org/) (for Comprehensive R Archive Network), and select a nearby mirror. For example, residents of New Zealand would be directed to <http://cran.stat.auckland.ac.nz/>

Precompiled distributions of R are available for free download for Windows, MAC OS X, and Linux, and installation is generally easy. For example, to install R for Windows, click on [Windows](#) > [Base](#) > [Download R for Windows](#). Then simply follow the download instructions. (For Windows users, use of the 32-bit version of R is presently recommended.) It will automatically install in the user's My Documents folder, but you may prefer to put it into the R library.

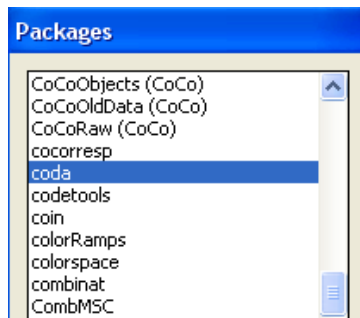
### 1b. Installing required R packages

The extension packages are available from the main CRAN page, but are more easily installed from R itself after it is installed.

- First, open R, and then select [Packages](#) > [Set CRAN mirror...](#) to select a nearby repository.
- Next, select [Install package\(s\)...](#)



- From here, select the desired packages (**R2WinBUGs**, **coda**, **lattice**, and **BRugs**), and click on **OK**. The packages will then self-install.



- Make sure all packages are installed in the R library (Program Files\R\R-X.X.X\library)

## 1c. Installing OpenBUGS

**OpenBUGS** is also free to install. **OpenBUGS** can be run as a stand-alone program, but will be interfaced through R in this tutorial. Therefore, while it must be installed, no knowledge of **OpenBUGS** will be required for the ISE calibration programs.

- Go to the **OpenBUGS** homepage (<http://www.openbugs.info/w/>). Windows and Linux users may download pre-compiled distributions by selecting **Download** from the **OpenBUGS** homepage

**OpenBUGS**

Search [ ] Titles Text

**Wiki**  
 Home Page  
 Overview  
 Downloads  
 Getting Started  
 Documentation  
 Developers  
 BugTracker  
 Community  
 Recent Changes  
 search  
 Site Navigation  
 How to Use This Site  
 Help Contents

**Page**  
 Immutable Page  
 Info  
 Attachments  
 More Actions: [ ]

**Downloads [what links here?]**

**OpenBUGS license**  
 Please click here, [License](#), before downloading OpenBUGS software

Release (3.2.1) of OpenBUGS is now available. Changes to the version 3.1.2 are documented in [changes](#). Release (3.2.1) of OpenBUGS has been tested and compared to the WinBUGS 1.4.3 program ([Test Results](#)). The only difference with WinBUGS 1.4.3 was with the 'Shared' GeoBUGS example. This discrepancy is still under evaluation. The computational speed of OpenBUGS is comparable to the WinBUGS ([@timing comparisons](#)). To report problems, please [register](#) and use the [BugTracker](#) page.

User contributed code, including **BRugs/R2OpenBUGS**, is located at [User Contributed Code](#).

**Microsoft Windows Download**  
 A setup program for OpenBUGS on Windows computers can be downloaded here: [@OpenBUGS321setup.exe](#).  
 The Windows program was tested on machines with the XP and Vista operating systems, and produced the same results.

**Macintosh Download**  
 The Windows version of OpenBUGS has been successfully installed on Mac OS X (Intel) using WINE. The Windows installer currently produces an error message at shutdown, but appears to install the program successfully. Testing under WINE has been VERY limited so far.

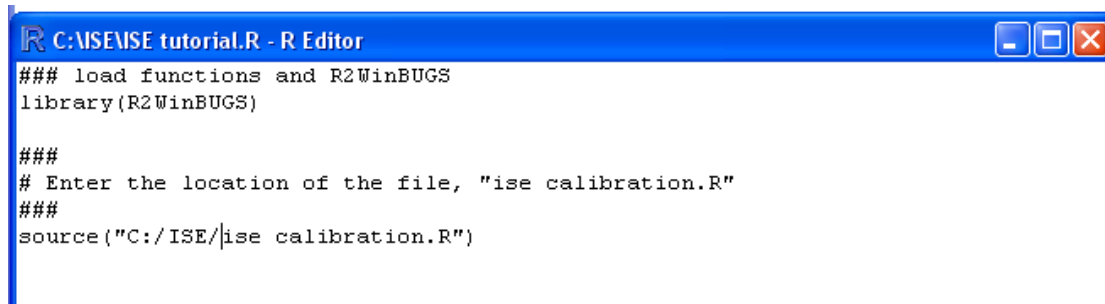
**Linux Download**

**Source package**  
 Should be usable on any x86 (PC) Linux platform. On 64 bit Linux, the necessary 32-bit C development packages are required. Compilation has been successful on 64-bit Ubuntu using the g++-multilib package, and on 64-bit Fedora using the glibc-devel package.  
 Download: [@OpenBUGS-3.2.1.tar.gz](#)

## 2. Analysing ISE data:

R is a powerful tool for statistical computing, and there are numerous tutorials and books about it. For ISE analysis, knowledge of script files is important and it is recommended you review a general R tutorial.

- Prior to analysing data, the R2WinBUGS library must be loaded by entering **library(R2WinBUGS)**
- An R script file called **ISE calibration.R** that contains ISE-specific functions was provided.



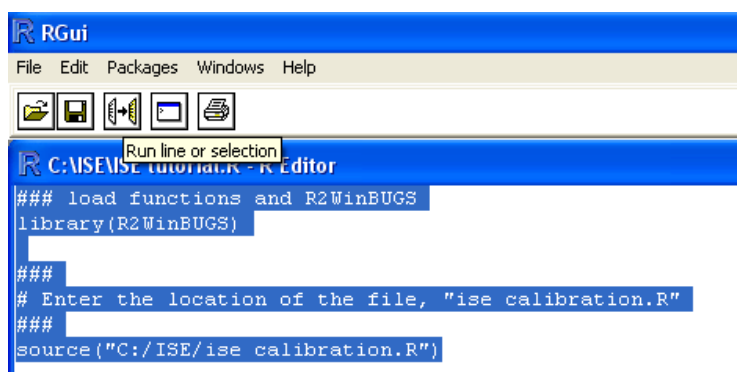
```
R C:\ISE\ tutorial.R - R Editor
### load functions and R2WinBUGS
library(R2WinBUGS)

###
# Enter the location of the file, "ise calibration.R"
###
source("C:/ISE/ise calibration.R")
```

- You may have saved the supplied file, “ise calibration.R” in a different location on your computer, and therefore the **pathname** ‘C:/ISE/ise calibration.R’ should be updated (e.g. enter **source(C:/correct path on your computer/ise calibration.R)**).

Note that R uses the ‘\’ character for other commands, and so pathnames are specified with a forward-slash (/) instead of the more common backwards slash (\).

- Running commands in a script file: At this point, no commands have been processed yet. In order to run the script file, highlight the commands you want to run, and click on the ‘Run line or selection’ icon. [Alternatively, Control-R runs either the current line if nothing is highlighted, or all highlighted commands in order.]



```
R RGui
File Edit Packages Windows Help
[Run line or selection icon]
R C:\ISE\ tutorial.R - R Editor
### load functions and R2WinBUGS
library(R2WinBUGS)

###
# Enter the location of the file, "ise calibration.R"
###
source("C:/ISE/ise calibration.R")
```

- The results of the commands appear in a separate window, called the R console; in this case, the console lets us know that the library and functions have been loaded.

```

R Console
> ### load functions and R2WinBUGS
> library(R2WinBUGS)
Loading required package: coda
Loading required package: lattice
>
> ###
> # Enter the location of the file, "ise calibration.R"
> ###
> source("C://ISE/ise calibration.R")
>

```

### 3a. Single ISE example:

The user only needs to interface with **two functions**, one to load data, and another to analyse the data. In the following example, there is one ISE, with 8 calibration measurements with known concentration. Additionally, there were seven experimental samples of unknown concentration. These datasets are available and named,

**ISE example data (calibration) ISE1.txt, and  
ISE example data (experimental) ISE1.txt**

Calibration Data			Experimental Data		
ISEID	log10x	emf	ISEID	SampleID	emf
1	-10	90.2	1	1	101.7
1	-9	90.2	1	2	110.1
1	-8	92.9	1	3	102.7
1	-7	97.4	1	4	150.4
1	-6	122.1	1	5	90.4
1	-5	152.9	1	6	121.2
1	-4	180.2	1	7	141.5
1	-3	207.9			

- The first function, **load.ISE.data**, requires that the user specify two filenames,
  - filename.experimental*: Location and name of the experimental data,
  - filename.calibration*: Location and name of the calibration data

Further, there is an additional variable that is used to indicate that the experimental data is standard addition data,

*stdadd*: Standard addition (T) or original model (F, the default)

**Formatting of data:** The data must be appropriately formatted, described in section 4, but the formatting is straightforward. Next, `load.ISE.data` will automatically detect whether there is a single ISE or multiple ISEs, and perform additional formatting. In this example, the final data object is stored as “`mydata.SingleISE`” (this name is arbitrary, see figure below).

- The second function, **bayesian.calibrate**, uses the calibration data to estimate the concentration for the experimental data, using the methods described in McGraw et al. (2011).

There are four required variables (`data`, `model.path`, `single.ISE`, and `Z`), and two experiment-specific optional variables (`stdadd`, `temperature`):

*data*: The name of the data object  
*model.path*: The location of the calibration model  
*single.ISE*: T or F, indicating data from a single (T) or multiple (F) ISEs  
*stdadd*: T or F, as defined for `load.ISE.data` (default is F)  
*Z*: The valence of the primary ion  
*temperature*: Defaults to 21C, if not entered

- Next, simply highlight the new code, run it, and wait a short time for the results:

```

R C:\ISE\ISE tutorial.R - R Editor
### load functions and R2WinBUGS
library(R2WinBUGS)

###
# Enter the location of the file, "ise calibration.R"
###
source("C:/ISE/ise calibration.R")

###
# Single ISE Example
###
# Load data
mydata.SingleISE = load.ISE.data(
  filename.experimental= "C:/ISE/ISE example data (experimental) ISE1.txt",
  filename.calibration= "C:/ISE/ISE example data (calibration) ISE1.txt")

# Run Bayesian analysis
example1 = bayesian.calibrate(mydata.SingleISE, model.path = "C:/ISE",
  model.name = "Single ISE model.txt", single.ISE=T, Z = 2)

```

- **Options:** There are additional options that do not need to be specified, but may be if the user desires:

### Graphical parameters:

Some customization of plots can be done by changing graphical parameters. More advanced users can also edit the function *bayesian.calibrate* in ISE calibration.R to customize graphics.

*plot.limits:* The range of the y axis for the sample concentrations plot

*x.ticks:* The location of tickmarks on the x axis (e.g. *x.ticks = c(1, 3, 5)* draws ticks at 1, 3, and 5)

*y.ticks:* The location of tickmarks on the y axis

### Technical MCMC inputs:

MCMC Estimation: This Bayesian analysis employs OpenBUGS, which implements Markov Chain Monte Carlo (MCMC) methods.

OpenBUGS implements a suite of MCMC algorithms. Starting from the initial set of parameter values, probabilistic decision rules are used to generate an updated set of parameter values. This process continues, generating a long chain of parameter values.

After a **burn-in** period where parameter values are not recorded, the parameter values are recorded for a given number of **iterations**. For diagnostic purposes, it is beneficial to have a number of parallel **chains**. In some cases, it is computationally efficient to run more iterations, but record fewer of them, done by **thinning** (e.g. *thin = 2* records every other iteration). The final parameter estimates are based on  $(iters - burnin) * chains / thin$  number of simulated parameter values.

The 100,000 parameter values from the default values of *burnin = 25000*, *iters = 50000*, *chains = 4*, and *thin = 1* are generally more than sufficient for good convergence.

*burnin:* The number of initial parameter estimates to discard while tuning the system

*iters:* The total number of iterations

*chains:* The number of parallel chains to run (useful for diagnostics)

*thin:* MCMC thinning rate, defaults to 1

*diagnostic.print:* Convergence diagnostic plots and printout

## Technical ISE inputs:

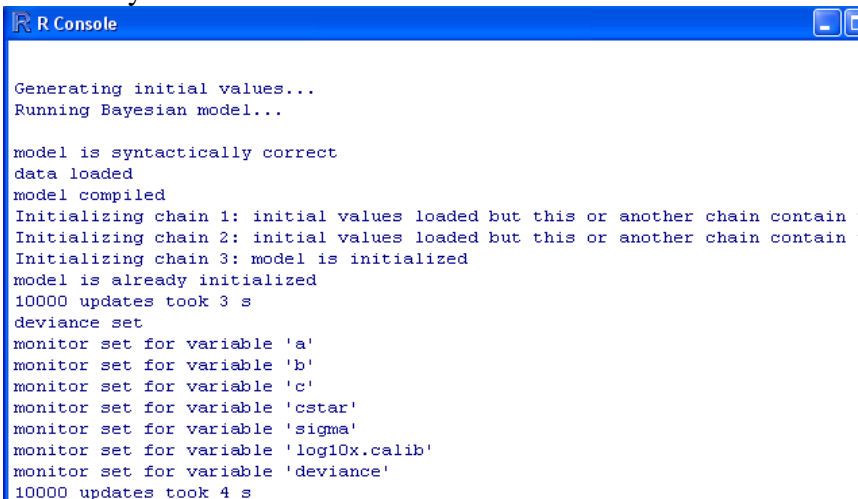
Initial values for MCMC calculations are automatically generated by the functions in ISE calibration.R. However, these automatically generated values occasionally produce poor initial values (particularly for the standard deviation,  $\sigma$ ).

- a.init*: Initial value passed to `gen.inits.single` or `gen.inits.multiple` for the model,  $\text{emf} = a + b \log(x + \text{cstar}^{10})$ . Default is NA. If there are **initial value out-of-bounds** errors, a value may be specified for a single ISE, or a vector may be specified for multiple ISEs; e.g. `a.init = 300`, or `a.init = c(300, 100, -50, 100)`
- b.init*: Similar to `a.init`
- cstar.init*: Similar to `a.init`
- sigma.upper*: Upper limit for initial value of sigma. A common error occurs when the **initial value for sigma** is too high. An appropriate value depends on the ISE and its application. For the challenging  $[\text{Pb}^{2+}]$  example in McGraw et al. 2011, `sigma.upper = 2 mV` works well.

## 3b. Single ISE Results:

In the example data set above, there were 7 samples with unknown concentrations measured by one ISE.

- The first portion of the output lets us know that the model compiled properly, and is only useful if an error occurs:



```
R Console
Generating initial values...
Running Bayesian model...

model is syntactically correct
data loaded
model compiled
Initializing chain 1: initial values loaded but this or another chain contain u
Initializing chain 2: initial values loaded but this or another chain contain u
Initializing chain 3: model is initialized
model is already initialized
10000 updates took 3 s
deviance set
monitor set for variable 'a'
monitor set for variable 'b'
monitor set for variable 'c'
monitor set for variable 'cstar'
monitor set for variable 'sigma'
monitor set for variable 'log10x.calib'
monitor set for variable 'deviance'
10000 updates took 4 s
```

- The next portion of the output provides median estimates and 95% credible intervals of the parameters from the calibration curve,  $emf = \hat{a} + \hat{b} \log(x + \hat{c})$  are given for the ISE, as well as the estimated standard deviation,  $\hat{\sigma}$ . Finally, the log limit of detection,  $\log LOD_{S/N=3}$ , is calculated from the median estimates of the other parameters:

```
R R Console
Non-linear parameter estimates and 95% CIs for
  y = a + b log(x + c)

ISE #1:
  Parameter estimate Lower limit Upper limit
a      2.97e+02      2.88e+02  3.08e+02
b      2.93e+01      2.75e+01  3.16e+01
c      8.78e-08      5.22e-08  1.56e-07
sigma  2.03e+00      1.12e+00  4.98e+00

Estimated (log) LOD (S/N=3): -7.27
```

For example, the estimated slope is  $\hat{b} = 29.3$ , but the slope could be between 27.5 and 31.6, while  $\log LOD_{S/N=3} = -7.27$ .

- Finally, median estimates (and the upper and lower limits of the 95% calibration interval) are given for each sample concentration:

```
R R Console
Bayesian calibration of experimental samples:

SampleID Estimated concentration Lower limit Upper limit
1      -6.91      -7.64      -6.61
2      -6.48      -6.79      -6.23
3      -6.85      -7.39      -6.56
4      -5.01      -5.21      -4.80
5      -8.33      -8.97      -7.44
6      -6.04      -6.28      -5.81
7      -5.31      -5.52      -5.10
```

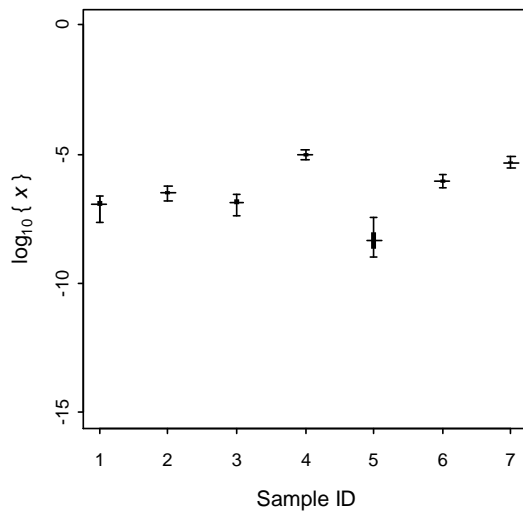
**Interpretation:** Estimates become more influenced by the assumed prior distribution of  $x$  for values below the detection limit, in this case estimated on a log scale as  $-7.27$ .

Estimated concentrations and lower confidence limits below  $LOD_{S/N=3}$  should be thought of as “**below LOD**”. For example, for the first unknown sample, the estimated concentration on a log scale is  $-6.91$ , and we are 95% confident that the true concentration is between “below LOD” and  $-6.61$ .

In this case, the data came from simulations, and so the true concentrations of the calibration data are known ( $-7.5, -6.5, -7.0, -5.0, -8.0, -6.0, -5.5$ ). In this example, all of the true concentrations are contained in the calibration intervals. In general, approximately 95% of sample concentrations will fall with their calibration interval if model assumptions are met (or are close to being met).



- A plot of the sample concentrations is also created in a separate window, and can be saved in a variety of formats. The default range on the y axis is large; this can be controlled using the `plot.limits` argument (e.g. `add, plot.limits = c(-10, -4)` in `bayesian.calibrate`):



### 3c. Multiple ISEs:

Analysing data from multiple ISEs is no more difficult than a single ISE. This data set has four ISEs measuring 8 calibration points and 7 unknown samples. Note that, while this data set is complete (every ISE measures every calibration and experimental point), this is not a requirement.

Calibration Data			Experimental Data		
ISEID	log10x	emf	ISEID	SampleID	emf
1	-10	90.2	1	1	101.7
1	-9	90.2	1	2	110.1
1	-8	92.9	1	3	102.7
1	-7	97.4	1	4	150.4
1	-6	122.1	1	5	90.4
1	-5	152.9	1	6	121.2
1	-4	180.2	1	7	141.5
1	-3	207.9	2	1	2.9
2	-10	-3.8	2	2	14.3
2	-9	-4.9	2	3	10.0
2	-8	-2.7	2	4	50.7
2	-7	5.7	2	5	-3.9
2	-6	23.5	2	6	17.9
2	-5	48.0	2	7	34.5
2	-4	82.9	3	1	-196.9
2	-3	105.6	3	2	-186.9
3	-10	-191.5	3	3	-188.8
3	-9	-197.7	3	4	-148.9
3	-8	-194.5	3	5	-192.3
3	-7	-191.5	3	6	-178.9
3	-6	-174.1	3	7	-165.6
3	-5	-146.8	4	1	-79.5
3	-4	-113.8	4	2	-79.4
3	-3	-86.7	4	3	-77.1
4	-10	-83.9	4	4	-46.7
4	-9	-78.0	4	5	-69.5
4	-8	-75.0	4	6	-67.2
4	-7	-78.3	4	7	-63.9
4	-6	-65.6			
4	-5	-50.8			
4	-4	-16.3			
4	-3	9.4			

- The code looks identical to the single ISE code, except that (1) different **data files** ('ISE example data (experimental).txt', and 'ISE example data (calibration).txt') are loaded that contain the multiple ISE experimental and calibration data, and (2) `single.ISE=T` changes to **single.ISE=F** in `bayesian.calibrate`:

```

R C:\ISE\ISE tutorial.R - R Editor
### Multiple ISE example
# Load data
mydata.MultipleISE = load.ISE.data(
  filename.experimental=
    "C:/ISE/ISE example data (experimental).txt",
  filename.calibration=
    "C:/ISE/ISE example data (calibration).txt")

# Run Bayesian analysis
example2 = bayesian.calibrate(mydata.MultipleISE, model.path = "C:/ISE",
  model.name = "Multiple ISE model.txt", single.ISE=F, Z = 2,
  plot.limits=c(-10, -4))

```

- The output is similar, except that parameter values are returned for each of the four ISEs.

```

R R Console
Non-linear parameter estimates and 95% CIs for
      y = a + b log(x + c)

ISE #1:
      Parameter estimate Lower limit Upper limit
a      2.97e+02           2.87e+02  3.07e+02
b      2.93e+01           2.72e+01  3.15e+01
c      8.78e-08           4.89e-08  1.53e-07
sigma  2.07e+00           1.12e+00  5.10e+00

Estimated (log) LOD (S/N=3): -7.26

ISE #2:
      Parameter estimate Lower limit Upper limit
a      1.92e+02           1.78e+02  2.12e+02
b      2.83e+01           2.52e+01  3.28e+01
c      1.19e-07           4.71e-08  3.42e-07
sigma  3.06e+00           1.67e+00  7.69e+00

Estimated (log) LOD (S/N=3): -6.88

ISE #3:
      Parameter estimate Lower limit Upper limit
a      6.34e+00           -8.77e+00  2.48e+01
b      3.06e+01           2.73e+01  3.48e+01
c      2.65e-07           1.17e-07  6.00e-07
sigma  2.83e+00           1.56e+00  6.90e+00

Estimated (log) LOD (S/N=3): -6.63

ISE #4:
      Parameter estimate Lower limit Upper limit
a      9.69e+01           7.01e+01  1.32e+02
b      2.90e+01           2.27e+01  3.83e+01
c      8.93e-07           1.82e-07  3.90e-06
sigma  4.75e+00           2.61e+00  8.98e+00

Estimated (log) LOD (S/N=3): -5.73

```

- Calibrated values: By using all of the ISEs, the precision of each estimated concentration improves. For example, with one ISE (the best performing ISE), the estimated concentration for the fourth sample was -5.0 (95% CI: -5.2, -4.8). Using all four ISEs, the 95% CI became 50 percent more precise: (-5.1, -4.9).

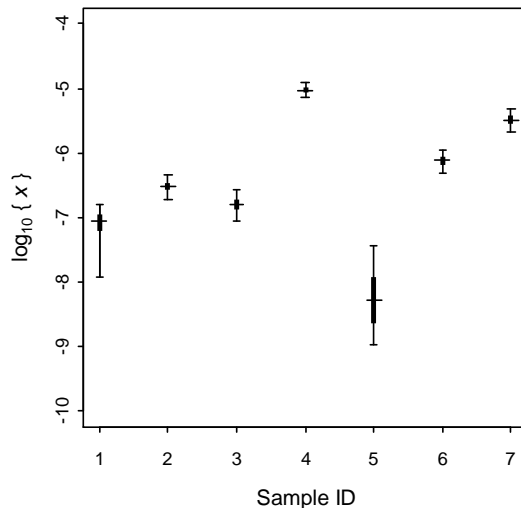
```

R R Console
Bayesian calibration of experimental samples:

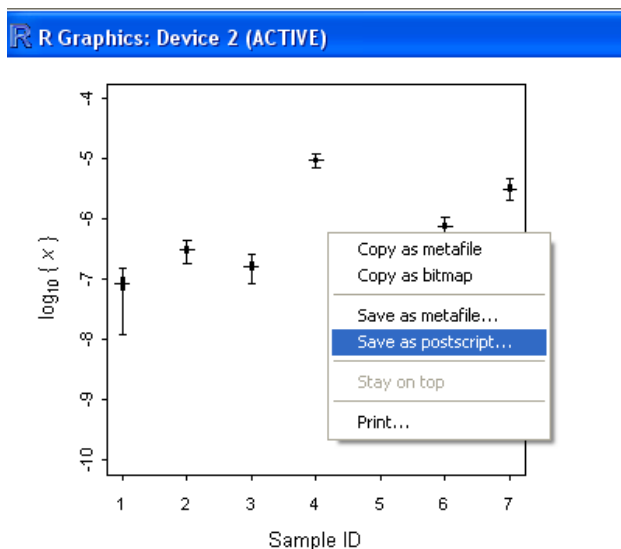
SampleID Estimated concentration Lower limit Upper limit
1         -7.06                 -7.92     -6.80
2         -6.51                 -6.73     -6.34
3         -6.79                 -7.06     -6.57
4         -5.02                 -5.14     -4.90
5         -8.29                 -8.96     -7.43
6         -6.12                 -6.30     -5.96
7         -5.49                 -5.68     -5.33

```

- Having used the plot.limits argument, an improved figure is created:



- A publication quality plot can be created by saving this plot in postscript format. Simply right-click, and then click on *Save as postscript...*



### 3d. Standard addition models:

When the experimental data is measured using the standard addition method, a modified model is used.

- In order to use the standard addition models that are supplied, the name of the file for the standard addition model must be specified (e.g. **model.name** = “**Single ISE model Standard addition.txt**” or **model.name** = “**Multiple ISE model Standard addition.txt**”), and
- **stdadd = T** must be specified in **load.ISE.data** and in **bayesian.calibrate**.

The example dataset used to demonstrate the standard addition model is the [Pb<sup>2+</sup>] dataset presented in McGraw et al. (2011).

```
R C:\SEISE tutorial.R - R Editor
###
# Standard addition example, Single ISE
###
mydata.SingleISE.sa = load.ISE.data(
  filename.experimental=
    "C:/ISE/ISE example data 2 (experimental standard addition) ISE1.txt",
  filename.calibration=
    "C:/ISE/ISE example data 2 (calibration standard addition) ISE1.txt",
  stdadd = T)

example3 = bayesian.calibrate(mydata.SingleISE.sa, model.path = "C:/ISE",
  model.name = "Single ISE model Standard addition.txt", stdadd = T,
  single.ISE=T, Z = 2, plot.limits = c(-7, -3))

###
# Standard addition example, Multiple ISEs
###
mydata.MultipleISE.sa = load.ISE.data(
  filename.experimental=
    "C:/ISE/ISE example data 2 (experimental standard addition).txt",
  filename.calibration=
    "C:/ISE/ISE example data 2 (calibration standard addition).txt",
  stdadd = T)

example4 = bayesian.calibrate(mydata.MultipleISE.sa, model.path = "C:/ISE",
  model.name = "Multiple ISE model Standard addition.txt", stdadd = T,
  single.ISE=F, Z = 2, plot.limits=c(-7, -3))
```

## 4. Formatting data:

Data needs to be in **two tab-delimited text files**, one for the calibration data, and one for the experimental data. There needs to be a header row for each file. Excel spreadsheets can be saved as tab-delimited text files (using Save As), or copied and pasted into text editors.

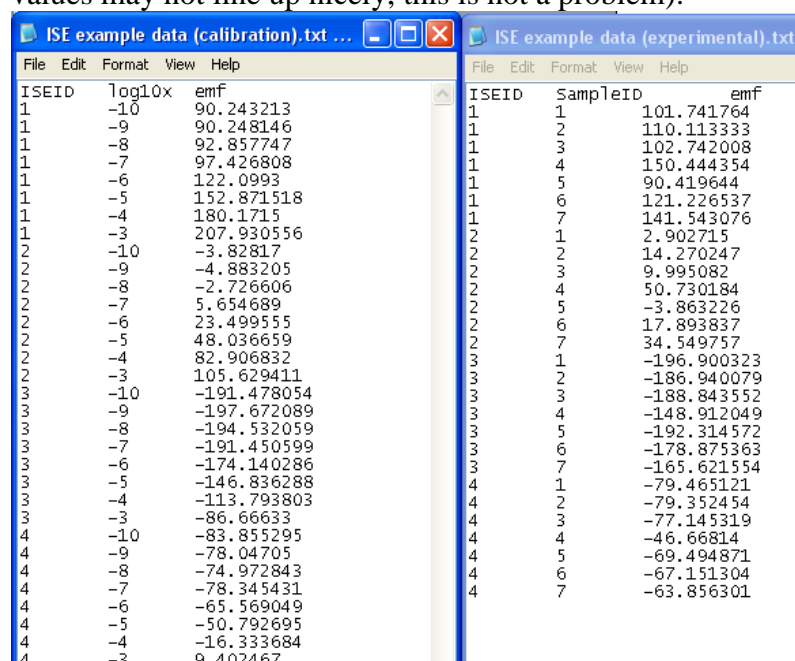
For the **calibration** file, the header rows are **ISEID**, **log10x**, and **emf** (the order does not matter). The ISEID variables should be integers, with the lowest value equal to 1, and no gaps. That is, if there are four ISEs, they must be labeled 1, 2, 3, and 4.

However, ISE #2 may come before ISE #1, or in mixed order. If there is only a single ISE, ISEID would equal 1 for all measurements. The next variable is log10x, the log concentration ( $\log_{10}$  mol/l) of the calibration samples. The emf readings (mV) follow.

The **experimental file** will take one of two formats, depending on whether the **original model** or the **standard addition model** is used.

- For the **original 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.
- 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 + 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.

Example files for the original model are pictured below (note that tab-delimited values may not line up nicely; this is not a problem):



The image shows two side-by-side screenshots of text editors. The left window is titled "ISE example data (calibration).txt ..." and contains a table with three columns: ISEID, log10x, and emf. The right window is titled "ISE example data (experimental).txt" and contains a table with three columns: ISEID, SampleID, and emf. Both tables contain multiple rows of data.

ISEID	log10x	emf
1	-10	90.243213
1	-9	90.248146
1	-8	92.857747
1	-7	97.426808
1	-6	122.0993
1	-5	152.871518
1	-4	180.1715
1	-3	207.930556
2	-10	-3.82817
2	-9	-4.883205
2	-8	-2.726606
2	-7	5.654689
2	-6	23.499555
2	-5	48.036659
2	-4	82.906832
2	-3	105.629411
3	-10	-191.478054
3	-9	-197.672089
3	-8	-194.532059
3	-7	-191.450599
3	-6	-174.140286
3	-5	-146.836288
3	-4	-113.793803
3	-3	-86.66633
4	-10	-83.855295
4	-9	-78.04705
4	-8	-74.972843
4	-7	-78.345431
4	-6	-65.569049
4	-5	-50.792695
4	-4	-16.333684
4	-3	9.402467

ISEID	SampleID	emf
1	1	101.741764
1	2	110.113333
1	3	102.742008
1	4	150.444354
1	5	90.419644
1	6	121.226537
1	7	141.543076
2	1	2.902715
2	2	14.270247
2	3	9.995082
2	4	50.730184
2	5	-3.863226
2	6	17.893837
2	7	34.549757
3	1	-196.900323
3	2	-186.940079
3	3	-188.843552
3	4	-148.912049
3	5	-192.314572
3	6	-178.875363
3	7	-165.621554
4	1	-79.465121
4	2	-79.352454
4	3	-77.145319
4	4	-46.66814
4	5	-69.494871
4	6	-67.151304
4	7	-63.856301

Finally, the  $[Pb^{2+}]$  calibration and experimental files for the standard addition method are below. Only the format of the experimental file changes.

The image shows two Notepad windows side-by-side. The left window, titled 'ISE example data 2 (calibration sta...', contains a calibration table with columns 'ISEID', 'log10x', and 'emf'. The right window, titled 'ISE example data 2 (experimental standard addition).txt - Notepad', contains an experimental data table with columns 'ISEID', 'sampleID', 'emf1', 'emf2', 'V. s', 'V. add', and 'conc. add'.

ISEID	log10x	emf
1	-9.00011625	8.558784008
1	-6.996850239	8.041667495
1	-5.962362208	13.82426611
1	-4.970695789	32.1609243
1	-3.996123497	56.68302162
1	-3.076334905	85.45835326
2	-9.00011625	-28.38121599
2	-6.996850239	-25.45833251
2	-5.962362208	-20.54573389
2	-4.970695789	-1.579075696
2	-3.996123497	24.77302162
2	-3.076334905	53.86835326
3	-9.00011625	-347.271216
3	-6.996850239	-320.7183325
3	-5.962362208	-298.3557339
3	-4.970695789	-268.2390757
3	-3.996123497	-237.1269784
3	-3.076334905	-204.0416467

ISEID	sampleID	emf1	emf2	V. s	V. add	conc. add
1	1	25.49	51.14	25	0.02	0.100
1	1	-7.69	19.31	25	0.02	0.100
1	1	-284.32	-246.38	25	0.02	0.100
1	2	29.61	52.78	25	0.02	0.100
1	2	-3.40	21.52	25	0.02	0.100
1	2	-274.83	-243.84	25	0.02	0.100
1	3	32.19	57.68	25	0.02	0.100
1	3	3.76	28.10	25	0.02	0.100
1	3	-283.80	-241.37	25	0.02	0.100
1	4	27.96	54.15	25	0.02	0.100
1	4	-3.30	23.30	25	0.02	0.100
1	4	-277.30	-243.24	25	0.02	0.100
1	5	22.39	37.95	25	0.25	0.001
1	5	-7.56	8.99	25	0.25	0.001
1	5	-295.17	-266.04	25	0.25	0.001
1	6	36.79	49.69	25	0.02	0.100
1	6	5.27	18.98	25	0.02	0.100
1	6	-256.82	-241.83	25	0.02	0.100
1	7	23.75	36.00	25	0.25	0.001
1	7	-5.46	6.20	25	0.25	0.001
1	7	-284.20	-264.51	25	0.25	0.001
1	8	10.55	30.32	25	0.25	0.001
1	8	-18.95	-0.46	25	0.25	0.001
1	8	-296.31	-267.46	25	0.25	0.001
1	9	70.58	88.78	25	0.25	0.100
1	9	41.59	60.20	25	0.25	0.100
1	9	-222.14	-199.14	25	0.25	0.100
1	10	20.42	34.22	25	0.25	0.001
1	10	-11.43	4.04	25	0.25	0.001
1	10	-289.36	-264.70	25	0.25	0.001
1	11	37.67	49.35	25	0.02	0.100
1	11	11.64	23.23	25	0.02	0.100
1	11	-274.11	-251.80	25	0.02	0.100
1	12	24.78	39.69	25	0.25	0.001
1	12	-4.38	10.95	25	0.25	0.001
1	12	-285.44	-260.25	25	0.25	0.001
1	13	20.98	35.74	25	0.25	0.001
1	13	-8.07	6.51	25	0.25	0.001
1	13	-290.61	-263.50	25	0.25	0.001
1	14	19.39	38.07	25	0.25	0.001
1	14	-9.02	7.72	25	0.25	0.001
1	14	-293.82	-262.97	25	0.25	0.001
1	15	21.33	38.43	25	0.25	0.001
1	15	-6.72	7.72	25	0.25	0.001
1	15	-295.30	-262.97	25	0.25	0.001
1	16	22.00	38.54	25	0.25	0.001
1	16	-6.19	8.85	25	0.25	0.001
1	16	-289.78	-261.89	25	0.25	0.001
1	17	41.23	56.97	25	0.02	0.100
1	17	16.40	31.00	25	0.02	0.100
1	17	-273.15	-244.69	25	0.02	0.100