Appendix

The data set used for this article is shown in the table. It is taken from Deming and Turoff (1978) where it was published in retention times. For this article, the data has been converted to their corresponding capacity factors.

pH	BA	OABA	PABA	HOBA
3.79	34.21	15.06	8.85	14.30
3.79	34.27	14.64	8.33	14.52
4.14	25.85	14.24	8.00	12.30
4.38	20.46	13.33	7.58	10.76
4.57	15.61	12.61	6.82	8.91
4.74	12.42	11.33	5.76	7.24
4.74	11.42	10.55	5.76	7.06
4.92	9.64	10.15	5.09	5.94
5.11	7.30	9.12	4.15	4.52
5.35	5.15	6.36	2.88	3.09
5.67	3.18	3.92	1.60	1.68
5.67	3.18	3.92	1.58	1.62

Fitting dose-response curves from bioassays and toxicity testing

by Johannes Ranke

Introduction

During the development of new chemicals, but also in risk assessment of existing chemicals, they have to be characterized concerning their potential to harm biological organisms. Characterizing chemicals according to this potential has many facettes and requires various types of experiments. One of the most important types is the dose-response experiment.

In such experiments, the responses of biological organisms to different doses[1] are observed in a quantitative way. Examples of the observed variables (endpoints of toxicity) are the length of wheat seedlings after being exposed to different concentrations of the chemical substance for a defined time interval, the activity of luminescent bacteria, the ability of cell cultures to reduce a specific dye, the growth rate according to number of individuals or biomass, the number of viable offspring and many others.

These observed variables have in common that a reference magnitude for healthy and viable organisms can be defined (normalised response level $r = 1$), and that the magnitude of the variable (response) is limited by a zero response ($r = 0$) where the maximum of the effect is observed. The **drfit** package covers the case where there is a continuum of possible response values between 0 and 1 (inclusive). Additionally, responses above 1 are frequently observed due to variability or as the result of stimulation by a subtoxic dose, and even responses below 0 may be present, depending on the type of data and the applied preprocessing.

If the responses are binomial, such as life and death for a number of individuals, it is adviseable to choose the readily available `glm` fitting procedures (generalized linear models), where the probit and logit links are already built-in (e.g. Chapter 7.2 in Venables and Ripley (2002)) or to look into the **drc** package.

Dose-response relationships for continuous response tests can generally be expressed as

$$r = f(d, \vec{p}) + \epsilon \quad (1)$$

where r is the normalised response at dose d, $f(d, \vec{p})$ is the model function with parameter vector $\vec{p}$, and ϵ is the error variable describing the variability in the observations not explainable by the model function $f(d, \vec{p})$.

This article shows how different model functions $f(d, \vec{p})$ can be conveniently fitted to such dose-response data using the R package **drfit**, yielding the vector of parameters $\vec{p}$ that gives the description of the data with the least residual error. The fitting can be carried out for many substances with a single call to the main function `drfit`.

The results that the user will probably be most interested in are the doses at which a response of 50 % relative to healthy control organisms is to be expected (termed ED_{50}), as this is a very robust parameter describing the toxicity of the substance toward the organism investigated.

The **drfit** package internally uses the R function `nls` for nonlinear regression analysis as detailed by Bates and Watts (1988). Confidence intervals for the model parameters are calculated by the `confint.nls` function from the **MASS** package as described in Venables and Ripley (2002).

drfit defines a dose-response data representation as a special case of an R dataframe, facilitates fitting standard dose-response models (probit, logit,

[1] The term dose is used here in a generalised way, referring to doses in the strict sense like mg oral intake per kg body weight as well as to measured concentrations in aquatic toxicity tests or nominal concentrations in cell culture assays.

weibull and linlogit at the time of this writing), and a function to produce different types of plots of the data as well as the fitted curves.

Optionally, the raw data can be kept in an external database connected by **RODBC**. This has proven to be useful if the data of a large number of dose-response experiments have to be evaluated, as for example in bioassays based on microtiter plates.

Recently, the R package **drc** containing similar functionalities to **drfit** has been uploaded to CRAN. Unfortunately, I have noticed the existence of this package only during the preparation of this article, after having maintained **drfit** on CRAN for almost one and a half years. Maybe in the future it will be possible to join forces.

In this introductory article, it is explained how the input data must be formatted, how dose-response curves are fitted to the data using the `drfit` function and in what ways the data and the models can be plotted by the `drplot` function. Since the package was actively developed during the preparation of this article, the reader is advised to upgrade to the latest `drfit` version available. Note that R $\geq$ 2.1.0 is needed for recent `drfit` versions.

Collecting dose-response data

The `drfit` function expects the dose-response data as a data frame containing at least a factor called 'substance', a vector called 'unit' containing the unit used for the dose, a column 'response' with the response values of the test system normalized using the "natural" zero response as 0, and the response of the control organisms as a "natural" 1. Therefore, values outside this interval, and especially values above 1 may occur in the normalized data. An example of such data can be easily obtained from the built-in dataset `XY`.

```
> library(drfit)
> data(XY)
> print(XY,digits=2)
   nr.   substance dose unit fronds response
1    1     Control    0 mg/L    174     1.050
2    2     Control    0 mg/L    143     0.973
3    3     Control    0 mg/L    143     0.973
4    4 Substance X   10 mg/L    147     0.983
5    5 Substance X   10 mg/L    148     0.986
6    6 Substance X   10 mg/L    148     0.986
7    7 Substance X  100 mg/L     63     0.651
8    8 Substance X  100 mg/L     65     0.663
9    9 Substance X  100 mg/L     55     0.598
10  10 Substance X  300 mg/L     20     0.201
11  11 Substance X  300 mg/L     22     0.238
12  12 Substance X  300 mg/L     25     0.288
13  13 Substance X 1000 mg/L     13     0.031
14  14 Substance X 1000 mg/L     16     0.113
15  15 Substance X 1000 mg/L     16     0.113
16  16     Control    0 mg/L    153     0.999
17  17     Control    0 mg/L    144     0.975
18  18     Control    0 mg/L    163     1.024
19  19 Substance Y   10 mg/L     20     0.201
20  20 Substance Y   10 mg/L     19     0.180
21  21 Substance Y   10 mg/L     21     0.220
22  22 Substance Y  100 mg/L     13     0.031
23  23 Substance Y  100 mg/L     12     0.000
24  24 Substance Y  100 mg/L     13     0.031
25  25 Substance Y  300 mg/L     12     0.000
26  26 Substance Y  300 mg/L     12     0.000
27  27 Substance Y  300 mg/L     14     0.061
28  28 Substance Y 1000 mg/L     12     0.000
29  29 Substance Y 1000 mg/L     12     0.000
30  30 Substance Y 1000 mg/L     12     0.000
```

Normalisation of the response data is not done within the **drfit** package. It can either be carried out with a typical spreadsheet file, with some extra lines of R code, or by an external procedure, while or before the data is read into a database.

If the data is collected and normalised using MS Excel, it can be easily transferred to R by saving it in CSV format, and reading it in using the R function `read.csv2` or alternatively by the `read.xls` function from the **gdata** package. If OpenOffice.org Calc is being used, and the default values are used for exporting the data in CSV format, the function `read.csv` is very helpful.

Figure 1 shows a possible spreadsheet layout for capturing dose-response data including both the observed endpoint (number of fronds in this case) and the normalized response values.

Total growth inhibition is in this case the natural lower limit of the response and the response will therefore be zero if the number of duckweed (*Lemna minor*) fronds stays at the initial level n_0 during the observation time. The natural reference for the healthy organisms (response=1) is in this case given by the growth rate of the controls μ_c, calculated by

$$\mu_c = \frac{\ln(\bar{n_c}) - \ln(n_0)}{t - t_0} \tag{2}$$

where $\bar{n_c}$ is the mean number of fronds in the control experiments after the observation time. The growth rates μ_i are calculated in the same way, and the normalized responses are then easily obtained by

$$r_i = \frac{\mu_i}{\mu_c} \tag{3}$$

If the spreadsheet from Figure 1 (which can be found at http://www.uft.uni-bremen.de/chemie/ranke/data/drfit/) were exported by writing a CSV file, this file could be processed by something like

```
> d <- read.csv('sampledata.csv',skip=2,dec=',')
```

depending on the path to the CSV file, the number of lines before the column headings and the decimal separator used.

F4 =(LN(E4)-LN(12))

	A	B	C	D	E	F
1	Concentration-response data for the Lemna growth tes					
2						
3	**nr.**	**substance**	**dose**	**unit**	**fronds**	**response**
4	1	Control	0	mg/L	174	1,0496
5	2	Control	0	mg/L	143	0,9726
6	3	Control	0	mg/L	143	0,9726
7	4	Substance X	10	mg/L	147	0,9834
8	5	Substance X	10	mg/L	148	0,9861
9	6	Substance X	10	mg/L	148	0,9861
10	7	Substance X	100	mg/L	63	0,6509
11	8	Substance X	100	mg/L	65	0,6631
12	9	Substance X	100	mg/L	55	0,5976
13	10	Substance X	300	mg/L	20	0,2005
14	11	Substance X	300	mg/L	22	0,2379
15	12	Substance X	300	mg/L	25	0,2881
16	13	Substance X	1000	mg/L	13	0,0314
17	14	Substance X	1000	mg/L	16	0,1129
18	15	Substance X	1000	mg/L	16	0,1129
19	16	Control	0	mg/L	153	0,9991
20	17	Control	0	mg/L	144	0,9754
21	18	Control	0	mg/L	163	1,0240
22	19	Substance Y	10	mg/L	20	0,2005
23	20	Substance Y	10	mg/L	19	0,1804
24	21	Substance Y	10	mg/L	21	0,2197
25	22	Substance Y	100	mg/L	13	0,0314
26	23	Substance Y	100	mg/L	12	0,0000
27	24	Substance Y	100	mg/L	13	0,0314
28	25	Substance Y	300	mg/L	12	0,0000

Figure 1: Data structure for a typical toxicity test in OpenOffice Calc. Note that the response column is calculated (see text).

Fitting and plotting

A quick result for a compatible dataframe can usually be obtained by a simple call to `drfit`

```
> rXY <- drfit(XY)
```

The contents of the dataframe `rXY` containing the results of the fitting procedure are shown in Figure 2. Each fitted dose-response model (usually only one per substance) produces one line. The number of dose levels `ndl` is reported, the total number of data points used for the model fitting `n`, the decadic logarithms of the lowest dose `lld` and the highest dose `lhd` tested.

The next column contains the type of the dose-response model fitted (probit, logit, weibull or linlogit) or, if not applicable, a classification of the substance data as "active" (if the response at the lowest dose is < 0.5) , "inactive" (if the response at the highest dose is > 0.5) or "no fit".

The log ED_{50} is given with its confidence interval as calculated by the `confint.nls` function from the **MASS** package. This only works if the log ED_{50} is one of the model parameters. Therefore, in the case of the weibull model, no confidence interval is given.

Finally, the residual sum of squares `sigma` is listed and the fitted parameters `a` and `b`, or, in the case of the three paramater model `linlogit`, the parameters `a`, `b` and `c` are listed.

Once the `drfit` function has been successfully called and the result assigned a name (`rXY` in this case), dose-response plots for the fitted data can easily be created using the `drplot` function. The following example produces a single graph (`overlay=TRUE`) with the fitted dose-response curves and raw data (`dtype="raw"`) for all substances and fitted models in dataframes `XY` and `rXY` using color (`bw=FALSE`). Additionally, the scatter of the responses in control experiments can be displayed, by setting the argument `ctype` to `"std"` or `"conf"`: as shown in Figure 3.

```
> drplot(rXY,XY,overlay=TRUE,bw=FALSE,
ylim=c("auto",1.3),dtype="raw", ctype="conf")
```

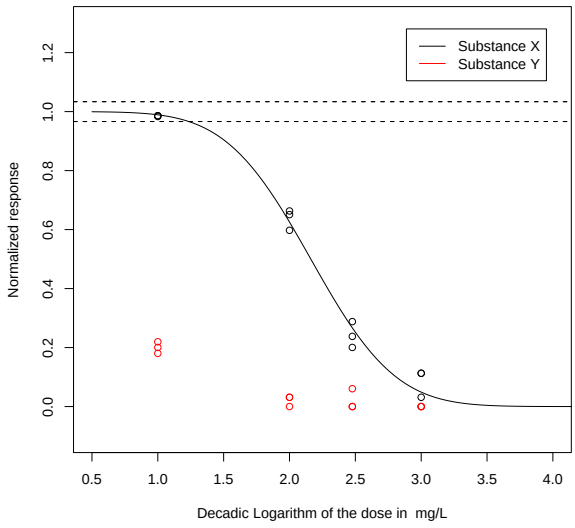

Figure 3: Output of the `drplot` function for the sample data `XY` from the package.

If the user prefers to view the raw data with error bars, the argument `dtype` can be set to `"std"` for showing standard deviations (default) or `"conf"` for showing confidence intervals.

In the following, the analysis of a somewhat more complicated, but also more interesting example is illustrated, which has been published by Ranke et al. (2004) before the basic `drfit` code was packaged .

First, dose-response curves with the default settings of `drfit` are generated as shown in Figure 4.

```
> data(IM1xIPC81)
> dIM <- IM1xIPC81
> rIM <- drfit(dIM)
> drplot(rIM,dIM,overlay=TRUE,bw=FALSE)
```

```
> print(rXY,digits=2)
    Substance ndl  n  lld  lhd    mtype logED50 2.5% 97.5% unit sigma   a    b
1     Control   1  6 -Inf -Inf inactive      NA   NA    NA mg/L    NA  NA   NA
2 Substance X   4 12    1    3   probit     2.2  2.1   2.2 mg/L 0.041 2.2 0.51
3 Substance Y   4 12    1    3   active      NA   NA    NA mg/L    NA  NA   NA
```

Figure 2: Contents of the dataframe containing the results from the fitting procedure for example data from the package (see text for explanations).

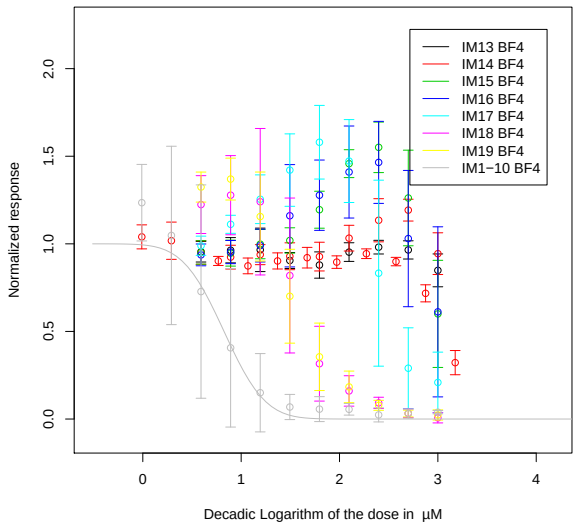

Figure 4: Dose-response plot showing the toxicities in a homologous series of compounds and the fitted probit model for IM1-10 BF4.

The graph shows that only one dose-response curve is fitted with the built-in default arguments of the `drfit` function and that the legend is interfering with the data. It is obvious that for almost all substances in this data, response values > 1 are caused in a certain dose range, a phenomenon which is called hormesis. In order to properly model such data, the so-called linear-logistic dose-response model has been introduced by Brain and Cousens (1989). The `drfit` package makes use of it in the parameterization suggested by van Ewijk and Hoekstra (1993), which allows for a convenient calculation of confidence intervals of the ED_{50}.

To include the linear-logistic model (`linlogit` in `drfit` terminology) in the fitting procedure and list the results including confidence intervals for a confidence level of 90 % two-sided, one simply calls

```
> rIM2 <- drfit(dIM,linlogit=TRUE,level=0.9,
chooseone=FALSE)
```

First, the `linlogit` argument causes the linlogit model to be additionally tried. Then, the argument `chooseone=FALSE` leads to reporting one line for each fitted model. If the argument `chooseone` is set to `TRUE` (default), only the first convergent dose-response model (probit and linlogit in this case) from the somewhat arbitrary sequence linlogit > probit > logit > weibull is reported.

The dataframe with the results shown in Figure 5 accordingly lists all instances of fitted models, and gives confidence intervals for the log ED_{50} values.

Then, a customized plot can be generated:

```
> drplot(rIM2,dIM,overlay=TRUE,bw=FALSE,
xlim=c("auto",5))
```

The `xlim` argument to `drplot` fixes the interference between legend and data. Furthermore, the plot produced in the above example shown in Figure 6 shows two fitted dose-response curves for the substance IM1-10 BF4 (grey lines), one for the probit and one for the linlogit model.

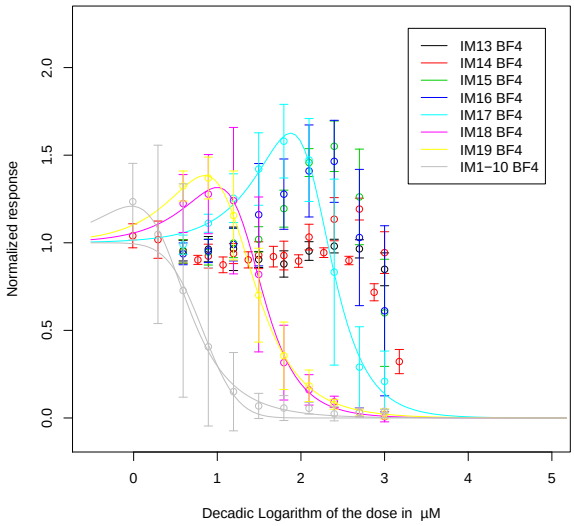

Figure 6: Dose-response plot showing the dose-response curves for a homologous series of compounds and all fitted linlogit and probit models.

External databases

Certain screening bioassays can be carried out with relatively low investments of time and money, so

```
> print(rIM2,digits=2)
   Substance ndl   n    lld lhd    mtype logED50   5%  95%   unit sigma    a    b     c
1   IM13 BF4   9  81  0.592 3.0 inactive      NA   NA   NA microM    NA   NA   NA    NA
2   IM14 BF4  20 216 -0.010 3.2   no fit      NA   NA   NA microM    NA   NA   NA    NA
3   IM15 BF4   9 135  0.592 3.0 inactive      NA   NA   NA microM    NA   NA   NA    NA
4   IM16 BF4   9 108  0.592 3.0 inactive      NA   NA   NA microM    NA   NA   NA    NA
5   IM17 BF4   9  81  0.592 3.0 linlogit    2.58 2.52 2.65 microM  0.24 2.58 2.30 0.015
6   IM18 BF4   9 135  0.592 3.0 linlogit    1.68 1.63 1.73 microM  0.23 1.68 2.24 0.057
7   IM19 BF4   9  81  0.592 3.0 linlogit    1.65 1.61 1.69 microM  0.15 1.65 1.98 0.110
8 IM1-10 BF4  11 162 -0.010 3.0 linlogit    0.77 0.70 0.84 microM  0.30 0.77 1.94 0.458
9 IM1-10 BF4  11 162 -0.010 3.0   probit    0.83 0.75 0.90 microM  0.31 0.83 0.33    NA
```

Figure 5: Contents of the dataframe containing the results from the fitting procedure for the chain length data `IM1xIPC81` from the package (see text for explanations).

large volumes of dose-response data can build up (high-throughput screening/high-content screening). The drfit package makes it possible to retrieve data stored in databases accessible by ODBC using the RODBC package internally. Since RODBC works on Windows, Mac OS X and Unix platforms, the code is platform- and database independent to a high degree.

For storing cytotoxicity data in a MySQL database, the following minimum database definition is advisable:

```
CREATE TABLE `cytotox` (
  `pk` int(11) unsigned NOT NULL auto_increment,
  `plate` int(11) NOT NULL default '0',
  `experimentator` varchar(40) NOT NULL
    default '',
  `substance` varchar(100) NOT NULL default '',
  `celltype` varchar(20) NOT NULL default '',
  `conc` float NOT NULL default '0',
  `unit` set('unit1','...') default 'unit1',
  `viability` float NOT NULL default '0',
  `performed` date NOT NULL
    default '0000-00-00',
  `ok` set('not ok','ok','?','no fit')
    default '?',
  PRIMARY KEY  (`pk`),
)
```

The `pk` and the `performed` data field are not interpreted by the package, databases with any other columns missing might work but have not been tested.

The column called `viability` contains the normalised response that has been calculated at the time of the data import into the database. Of course, the Data Source has to be configured to be a valid and accessible ODBC DSN on the platform used, e.g. by installing and configuring unixodbc and myodbc under Linux or MyODBC under Windows. This also involves setting up the MySQL server to listen to network connections, if it is not located on the local computer, and adequate MySQL user privileges.

With such a setup, the `drdata` function from the package can be used to conveniently retrieve data from the database and evaluate it with the `drfit` and `drplot` functions:

```
> s <- c("Sea-Nine","TBT","ZnPT2")
> d <- drdata(s,experimentator = "fstock",
whereClause="performed < 2006-04-04")
> r <- drfit(d,linlogit=TRUE)
> drplot(r,d,dtype="none",
bw=FALSE,overlay=TRUE)
```

The `whereClause` argument to the `drdata` function allows for flexible selection of data to be used for the analysis, e.g. by using comparison operators on columns containing dates as illustrated in the above example.

Additionally, the use of the argument `dtype="none"` to the `drplot` function is shown, which leads to the display of the fitted models only, without the data, as shown in Figure 7.

In the UFT Center of Environmental Research and Technology, we use the `drfit` package for regular batch-processing of all our dose-response data from several bioassays for a substance library of more than 200 compounds. The results are in turn written to a database, and the `drplot` function is used to create updated dose-response plots every time the raw data has been augmented. The whole process of fitting all data and producing the plots takes less about 1 minute on an 1600 MHz AMD Sempron PC for the cytotoxicity data for 227 substances, provided that the new data has been checked by the `checkplate` and `checksubstance` functions, which allow for an easy validation of experimental dose-response data generated by plate-reader bioassays stored in a `drfit` conformant MySQL database.

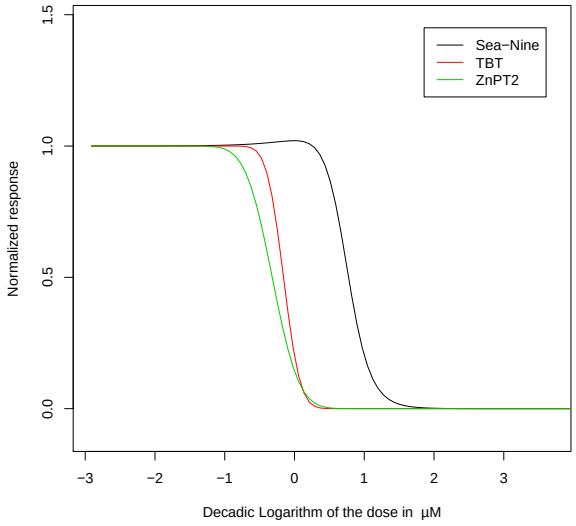

Figure 7: Dose-response plot showing the fitted dose-response curves for three antifouling biocides in the cytotoxicity assay fitted with the linlogit model.

The whole system provides the basis for analysis of the toxicity data, e.g. by (Quantitative) Structure-Activity Relationships (SAR/QSAR), which may provide a deeper chemical understanding of the interaction of the chemicals with biological organisms.

Bibliography

D. M. Bates and D. G. Watts. *Nonlinear Regression Analysis and its Applications*. Wiley Series in Probability and Mathematical Statistics. Wiley, New York, 1988. 7

P. Brain and R. Cousens. An equation to describe dose responses where there is stimulation of growth at low doses. *Weed Research*, 29:93–96, 1989. 10

J. Ranke, K. Mölter, F. Stock, U. Bottin-Weber, J. Poczobutt, J. Hoffmann, B. Ondruschka, J. Filser, and B. Jastorff. Biological effects of imidazolium ionic liquids with varying chain lengths in acute Vibrio fischeri and WST-1 cell viability assays. *Ecotoxicology and Environmental Safety*, 28(3):396–404, 2004. 9

P. H. van Ewijk and J. A. Hoekstra. Calculation of the EC50 and its confidence interval when subtoxic stimulus is present. *Ecotoxicology and Environmental Safety*, 25:25–32, 1993. 10

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Statistics and Computing. Springer, New York, 2002. 7

Johannes Ranke
Department of Bioorganic Chemistry
UFT Center for Environmental Research and Technology
University of Bremen jranke@uni-bremen.de

The pls package

by Bjørn-Helge Mevik

Introduction

The **pls** package implements *Partial Least Squares Regression* (PLSR) and *Principal Component Regression* (PCR). It is written by Ron Wehrens and Bjørn-Helge Mevik.

PCR is probably well-known to most statisticians. It consists of a linear regression of one or more responses **Y** onto a number of principal component scores from a predictor matrix **X** (Næs and Martens, 1988).

PLSR is also a linear regression onto a number of components from **X**, but whereas principal component analysis maximizes the variance of the scores, PLS maximizes the covariance between the scores and the response. The idea is that this should give components that are more relevant for the response. Typically, PLSR achieves the same (or smaller) prediction error as PCR, with fewer components. A good introduction to PLSR and PCR can be found in Martens and Næs (1989). A review of PLSR is given in Wold et al. (2001) (in fact, all of that issue of Chemolab is dedicated to PLSR). Frank and Friedman (1993) provides a more technical treatment, from a statistical viewpoint.

PLSR and PCR are commonly used in situations where there are collinearities or near-collinearities in **X**, for instance when there are more variables than observations. This is a very common situation in fields like chemometrics, where various types of spectroscopic data are often used to predict other measurements.

There are other regression methods that can be applied to such data, for instance ridge regression. Studies have indicated that in terms of prediction error, ridge regression can perform slightly better than PLSR. However, one of the major advantages of PLSR and PCR is interpretation. In addition to a prediction equation, one gets score and loading vectors