SDAR
Serial Data Analysis and Regression

version 5.2

Table of Contents

1. Summary

SDAR (Serial Data Analysis and Regression) is a Java application for graphical analysis, transformation and fitting of two-dimensional data. Numeric data in the form of multi-column ASCII files can be read into the program and graphed as *x*-*y*-plots. Data are output in a two-column format compatible with Grace, and sessions can be saved. Images of the **`Graph`** panel can be output in SVG, PNG or TIFF format. When using SDAR, please cite [1].

2. General layout and operation

SDAR uses tabbed panels to enable viewing of datasets. The main panel tabbed **`Graph`** shows graphical *x*-*y*-plots of the current datasets. For each dataset, a new tabbed panel is added with the name of the set showing as label in the tab. These latter panels show the spreadsheet format of the dataset, comprising of the *x*-*y*-data in the first columns, as well as any data derived from analysis in SDAR in the following columns. At the bottom of these panels, two functions are provided: **`Close`** will delete this dataset from the current session, **`Save`** writes the current dataset to an ASCII file compatible with the format of the program Grace; data derived from analysis in SDAR will be saved as remarks (indicated by **`#`**) at the top of the file. In the table view, the user can change data entries in the first two columns. With **`Update`** the amended data get plotted in the **`Graph`** panel (note that in order to save the amended data, the **`Save`** option still needs to be executed).

A movable tool bar allows quick access to frequently used graphics functions.

The current session can be saved in a binary file in SDAR format using the **`File Save`**, and retrieved with the **`File Open`** function.

On the **`Graph`** panel, vertical and horizontal line cursors can be activated and positioned using either by click with the **`left mouse button`**, or the arrow keys **`UP`**, **`DOWN`**, **`LEFT`** and **`RIGHT`**. The *x*- and *y*-values of the active dataset (see 4.1) at the current cursor position are displayed in the status bar. Transformations can be applied simultaneously to any selection of datasets. The position of the legend, as well as the position of the outer boundaries of the coordinate system can be changed by click and drag with the **`left mouse button`**.

This software is part of the Program Collection for Structural Biology and Biophysical Chemistry (PCSB) [2]; non-PCSB libraries used by this software are listed in section 7. This version of SDAR was compiled with Java 1.7.

2.1 How to load

Data can be loaded into SDAR

- with the **`File Open`** option to import *x*-*y*-data using a smart parser to find numerical data organised into columns in an ASCII file.
- with the **`File Open`** option to load a previous SDAR session.
- by starting the program with a data file name as argument in the command line
- copy-pasting or entering data manually using the **`Edit Add dataset`** option.

The **`File Open`** option can read SDAR files in ASCII and binary format (the new proprietary format since version 2.1); the program will automatically determine which file format is presented. SDAR files or ASCII data files can also be loaded on startup from the terminal by using the **`-i`** option:

```
java -jar sdar.jar -i {filename}
```

If the file is the only argument, the `-i` switch can be omitted:

```
java -jar sdar.jar {filename}
```

ASCII files are parsed using a 'smart parser' that attempts to find a single continuous data block in the file presented and determine the type of data separation (space, comma, etc.). If the file presented is in Grace format, a Grace parser is invoked that can load multiple datasets and interpret many of the meta commands.

2.2 How to save

Data processed in SDAR can be saved in different ways:

- Data can be saved using the **`File Save`** option as an ASCII file in binary SDAR-format (the *x*-*y*-data are fully contained in this file, no separate files with individual dataseries are generated). This option will save all current dataseries at once. Note: since version 2.1, these SDAR files are saved as binary files.
- Data can be written out as an ASCII file fully compatible with Grace using the **`File Export to Grace`** option. This option will save all current dataseries at once.
- Individual dataseries can be saved as ASCII files by using the **`Save`** option on the individual spreadsheet panes. These files will have remarks (indicated by `#`) with information from SDAR at the top. The remark section is followed by instructions for Grace (indicated by `@`). The *x*-*y*-data are then listed in two-column format.

2.3 Starting the software from terminal

A few options are available when starting SDAR from the terminal:

`java -jar sdar.jar`	starts the program
`java -jar sdar.jar {data file}`	starts the program and imports data
`java -jar sdar.jar [options]`	starts the program with the following options
`-debug`	be verbose
`-h`	print this information
`-i {filename}`	load ASCII data or SDAR binary file on startup
`-n {number}`	max. number of datasets

3. Description of user interface menus
3.1 File
Open
Data from ASCII data files with multiple columns can be imported with this function. The parser will analyse the ASCII file as to the presence of a continuous section of a certain number of columns. If there are more than two columns, the user can select which column shall be assigned to `X`, `Y` or `Sigma`. The data type can be chosen as `Continuous` (displayed as line plot in the `Graph` panel), `Discrete` (displayed as unconnected symbols in the `Graph` panel) or `Bar` (displayed as rectangular bars in the `Graph` panel).
If the data file to be loaded has consecutive multiple *y*-values for the same *x*-value, the *y*-values are averaged and only one *x*-*y*-data pair is passed on to SDAR.
A previous session saved with SDAR (SDAR binary file) can also be loaded with this function.

Save
The currently active session can be saved into an SDAR binary file.

Export to Grace
Data can be written out as an ASCII file fully compatible with Grace using the `File Export to Grace` option. This option will save all current dataseries at once. Relevant instructions for the Grace program will be at the top of the file (indicated by `@`). Bar graphs are currently not supported by this feature. If plot titles, axes titles or legends have been entered with basic HTML formatting tags or Greek symbols (e.g. `α`), they will be translated to Grace formatting.

Save image
The `Graph` panel can be saved as an image in the formats SVG, PNG or TIF. In order to generate scalable vector graphics (SVG) images in SDAR, the Apache Batik SVG toolkit has been implemented. Other image formats are then generated from the initial SVG image by transcoders within Batik.

Exit
This exits the program; changes to datasets the user has not saved will be lost.

3.2 Edit
Copy
This function copies the current content of the `Graph` panel into the clipboard.

Add dataset
This opens a window and allows copy-pasting of data from spreadsheet programs. Data can also be entered manually. In the latter case, values entered in each cell of the table need to be "verified" by hitting the `Enter` key (moving through the table with the `arrow keys` may not register the values correctly).

Change order of datasets
This function opens a window presenting the current datasets as an interactive list. By clicking `OK`, the order of current list is applied to the order of datasets in current session.

3.3 Format
View size
Sets the size of the `Graph` panel; default: 1075 × 526.

Plot properties
This allows setting the title and subtitle of the plot (basic HTML formatting tags, including Greek symbols, are supported), including font type, size and style.

Axes properties
The titles of the *x*- and *y*-axis can be entered (basic HTML formatting tags, including Greek symbols, are supported), and properties of the scales and tick intervals for both axes can be set. The position of the left and right *y*-axes as well as the lower and upper *x*-axes can be changed by pressing the **left mouse button** at the respective axis and dragging it to its new location.

Show grid
Turns background grid on or off.

Show legend
If legend labels for the current datasets have been entered by the user, the legend box will be shown or hidden on **Graph** panel. Legend labels can be entered using the Dataset button from the tool bar (see 4.1). The legend box can be re-positioned by pressing the **left mouse button** inside the legend box and dragging it to its new location.

3.4 Transform X

Translate X

With this function, a translation of the datasets activated for transformation (see **Transform** in 4.1) along the *x*-axis can be achieved. This is may e.g. be useful to re-define the point of $x = 0$ in a dataset.
A pop-up window allows the user to enter which new value (**New**) a current *x*-value (**Old**) should adopt; the numbers can be entered into the text fields. Alternatively, the vertical *x*-cursor is active and can be positioned by the user to indicate the **Old** *x*-value that should adopt the value specified in **New**.
The transformation applied to the datasets is: $x_{new} = x_{old} + (\mathbf{New} - \mathbf{Old})$

Scale X

This allows the user to multiply a constant scale factor into all current *x* values; the **scale** factor is entered into a text field in a pop-up window. This applies to all datasets currently activated for transformation.
The transformation applied to the dataset is: $x_{new} = x_{old} \cdot \mathbf{scale}$

Change data pitch

This allows the user to change the data pitch of the currently active datasets. *x*-*y*-pairs that have a smaller Δx than the new data pitch to their preceding *x*-*y*-pair will be deleted. This applies to all datasets currently activated for transformation.

3.5 Transform Y

Translate Y

With this function, a translation of the datasets activated for transformation along the *y*-axis can be achieved. A pop-up window allows the user to enter which new value (**New**) a current *y*-value (**Old**) should adopt; the numbers can be entered into the text fields. Alternatively, the horizontal *y*-cursor is active and can be positioned by the user to indicate the **Old** *y*-value that should adopt the value specified in **New**.
The transformation applied to the datasets is: $y_{new} = y_{old} + (\mathbf{New} - \mathbf{Old})$

Scale Y

This allows the user to multiply a constant scale factor into all current *y* values; the **scale** factor is entered into a text field in a pop-up window. This applies to all datasets currently activated for transformation.
The transformation applied to the dataset is: $y_{new} = y_{old} \cdot \mathtt{scale}$

Baseline linear

This function subtracts a linear function from all datasets activated for transformation. This is useful for example where baseline or drift corrections need to be carried out. A line with two squares at both ends will appear on the **Graph** panel and can be modified by the user as to the slope and *y*-position. To change the slope, the user needs to move either square up or down using the **left mouse button**. To change the *y*-position of the entire line, a vertical movement outside the squares is required.
The transformation applied to the dataset is: $y_{new} = y_{old} - (m \cdot x + t)$, where *m* and *t* are determined from the line modified graphically by the user.

Dataset addition

Here, graph arithmetic can be performed, i.e. pairwise addition or subtraction of datasets. Using selected entries from two drop down lists, the second dataset will be added to the first dataset with a multiplication factor added in the text field between. SDAR will produce the resulting dataset based on the *x*-values and data pitch of the first dataset within the shared *x*-region of both datasets. *y*-values for the second dataset are calculated by cubic spline interpolation.

Smooth

Currently, three smoothing types are implemented: **Moving mean** window, **Moving median** window, and **Savitzky-Golay** filter. These functions will open a pop-up window where the user can specify the degree of smoothing using a slider. The sliding mean/median window algorithms perform local averaging of an equally spaced function using the local mean or median within the window of neighbouring points determined by the slider position. Savitzky-Golay filtering is a convolution method, whereby successive sets of adjacent data points are fitted with a low-degree polynomial. The smoothing applies to all datasets currently activated for transformation.

3.6 Analysis

Maxima

With this menu item, an automatic determination of maxima in the currently active dataset will be performed. The maxima found by SDAR are highlighted in the **Graph** panel by blue vertical lines, and listed in pop-up window next to tick boxes. The user can accept or deny individual maxima by ticking/unticking. Further maxima can also be added manually, by clicking on **Add Peak**; this will activate the green vertical *x*-cursor that can be placed the user at the appropriate position. Hitting **ENTER** will include the current *x*-value in the list of maxima.
This procedure is required prior to attempt **Curve fitting** with **Gaussian** functions.

Minima

Minima determination is performed as described in section 3.6 for maxima.

Integration

Data can be integrated within a selected interval. A pop-up window will open where the user can enter the start (**From X**) and end (**To X**) of the interval. Alternatively, pressing and dragging the left mouse button, the area to be integrated can be highlighted graphically in the **Graph** panel. Hitting **ENTER** or clicking **OK** will perform the integration. The algorithm used by SDAR is based on numerical integration by Simpson's rule.

3.7 Curve fitting

For curve fitting, the user can choose between linear regression, non-linear Melder-Nead Simplex or Levenberg-Marquardt minimisation, depending on the type of equation. All curves can also be fitted manually using the sliders provided next to each parameter in the **Manual Fit** panel. If multiple datasets are loaded, the dataset to be fitted needs to be selected as **Active** under **Datasets** (section 4.1).

In the following cases, a pop-up window will open that allows the user to enter the fitting area (**Fit from X**, **to X**) and initial values for the fitting parameters. SDAR will provide an initial guess for all parameters. Individual parameters can be kept fixed during the minimisation calculation by unticking the **Fit** checkbox.

In addition to four goodness-of-fit parameters (see section 6), standard errors for the individual fit parameters are reported.

Burst Kinetics	$c(\mathrm{Product}) = A \cdot (1 - \mathrm{e}^{-k_b \cdot t}) + v_{ss} \cdot t$	This will fit an equation used to model burst kinetics, using either non-linear Simplex, Levenberg-Marquardt minimisation or manual fit. *A* is the burst amplitude, k_b the rate constant for the burst phase and v_{ss} the rate for the steady state.
Dose Response (EC_{50})	$y = \mathrm{Bottom} + \frac{\mathrm{Top} - \mathrm{Bottom}}{1 + \left(\frac{x}{\mathrm{EC}_{50}}\right)^{-\mathrm{Width}}}$	This will fit a logistic EC_{50} equation, using either Levenberg-Marquardt minimisation or manual fit.
Dose Response (IC_{50})	$y = \mathrm{Top} + \frac{\mathrm{Bottom} - \mathrm{Top}}{1 + \left(\frac{x}{\mathrm{IC}_{50}}\right)^{-\mathrm{Width}}}$	This will fit a logistic IC_{50} equation, using either Levenberg-Marquardt minimisation or manual fit.
Exponential	$y = a \cdot e^{-\lambda \cdot x} + b$	This will fit an exponential equation, using either non-linear Simplex, Levenberg-Marquardt minimisation or manual fit.
Gaussian	$y = \Sigma_i \, \mathrm{Amplitude}_i \cdot e^{-\frac{x - \mathrm{Centre}_i}{\mathrm{Width}_i}}$	Fitting of one or multiple Gaussian functions requires the prior definition of maxima (section 3.6). Then, the sum of Gaussians can be fitted using the equation above. The user can choose between either non-linear Simplex, Levenberg-Marquardt minimisation or manual fit.
Hill	$y = \frac{A \cdot x^n}{k^n + x^n}$	This will fit a Hill equation, using either non-linear Simplex, Levenberg-Marquardt minimisation or manual fit.
Hill with background	$y = \frac{A \cdot x^n}{k^n + x^n} + (m \cdot x + t)$	This will fit a Hill equation with optional linear background, using either non-linear Simplex, Levenberg-Marquardt minimisation or manual fit.
Langmuir	$y = \frac{K \cdot \mathrm{Top} \cdot x}{K \cdot x + 1}$	This will fit a Langmuir equation where *top* is maximum asymptotic value, using either non-linear Simplex, Levenberg-Marquardt minimisation or manual fit.

Linear	$y = m \cdot x + t$	This will fit a linear equation, using either simple linear regression (data without *sigma*), linear regression, Levenberg-Marquardt minimisation or manual fit.
Michaelis-Menten	$v = \frac{k_2 \cdot c_0(\mathrm{S}) \cdot c_0(\mathrm{E})}{K_\mathrm{M} + c_0(\mathrm{S})}$	This will fit the Michaelis-Menten equation, using either non-linear Simplex, Levenberg-Marquardt minimisation or manual fit.
Sigmoid	$y = \frac{\mathrm{Amplitude}}{1 + \mathrm{e}^{-\frac{x - \mathrm{Inflection}}{\mathrm{Width}}}} + \mathrm{Intercept}$	This will fit a sigmoid equation, using either non-linear Simplex, Levenberg-Marquardt minimisation or manual fit.

Manual (Gaussian)
Gaussian functions can be fitted to data graphically by performing a `right-click` with the mouse where the centre of the Gaussian should be. The centre can then be moved by pressing the left mouse button on the maximum of the displayed curve and dragging it to desired position; vertical dragging will increase the amplitude. Pressing the left button of the mouse while pointing to either one of the two half-maxima of the curve will widen the Gaussian. A pop-up window will inform about the current parameters of the fitted Gaussians and the fit statistics to the data.

3.8 Secondary plot

These functions will generate new graphs from currently loaded datsets. All current dataset spreadsheets will automatically be closed (note: unsaved data will be lost). New spreadsheets depending on the selected features below will be generated and populated with values accordingly.

1/y vs 1/x
All datasets selected to be transformed in the `Datasets` tool will be transformed into x^{-1} and y^{-1} and plotted in the `Graph` panel. Data points where either x or y equals 0 will be omitted. Sigma values will also be omitted from the transformation.

3.9 Settings

Function Resolution
Default: 100
This is the number of discrete data points that are calculated by SDAR when calculating the values of functions to be displayed.

Screen Resolution
Default: 90 dpi.
The screen resolution is required for calculating the size of output bitmap images using the specified `Output Image Resolution`. The default of 90 dpi is an estimated value; the user can update this estimate with the correct resolution used by their monitor.

Output Image Resolution
Default: 600 dpi.
Here, the user can set the resolution of the output bitmap image.

3.10 Help

Check For Update

Queries our web site to see if a new version of the software is available.

License Terms

Displays the license terms for this software.

About

Displays the program version.

4. Description of the toolbar

4.1 Datasets

This button will open a pop-up window with a list of currently loaded datasets. One of the datasets can be selected as `Active` and enable evaluation of cursor values displayed in the status bar underneath the `Graph` panel. Setting a dataset `Active` is also required to select it for curve fitting.

Multiple datasets can be selected for transformation operations by ticking the `Transformation` box next to each individual dataset. Under `Properties`, buttons with the current graph colour are being displayed. Clicking on these buttons will open another window that allow the user to change the dataset name, colour, line, symbol or bar settings, and enter a description for the legend (basic HTML formatting tags, including Greek symbols, are supported). Under the heading `Graph Type` a drop down menu allows changing between `Continuous`, `Discrete` and `Bar` graphs.

4.2 Autoscale

This button will auto-scale the plots in the `Graph` panel to display all data with respect to *x* and *y*.

4.3 Zoom

After clicking this button, the user can define a rectangle (`left mouse button drag`) on the `Graph` panel which will re-define the boundaries of the plot area to be shown.

4.4 X Cursor

This button activates the vertical line cursor that can be positioned with a click on the `left mouse button` or the `LEFT ARROW / RIGHT ARROW` keys. In the status bar, the current *x*- and *y*-values of the `Active` dataset will be displayed. A second click on this button will deactivate the *x*-cursor.

4.5 Y Cursor

This button activates the horizontal line cursor that can be positioned with a click on the `left mouse button` or the `UP ARROW / DOWN ARROW` keys. Where the *y*-cursor intersects with the currently active dataset, vertical drop lines are displayed. In the status bar, the current *y*-value will be displayed. A second click on this button will deactivate the *y*-cursor.

4.6 Mask

When this function is activated, the user can mask data points enclosed in a rectangular area. The area is defined by a `left mouse button drag` on the `Graph` panel; all data points of `Active` datasets enclosed in the rectangular area are flagged as 'masked'. In the `Graph` panel, masked data points appear in lighter colour; in the table representation of the datasets, masked points appear are highlighted by a grey background. Masked datapoints are not included in fitting procedures. If this operation is applied to datapoints that are currently masked, these points will be unmasked.

5. Dataset panels

Each dataset panel displays a table with a structured layout. The first three columns hold *x*-, *y*- and σ-values, respectively, and each cell can be edited. Editing of these values will update the data on the `Graph` panel (note: editing will not prompt a new analysis of maxima, minima, etc). Deletion of either an *x*- or a *y*-value will result in deletion of this data point. Deletion of a σ-value will set this value to 0.

6. Statistical parameters

Fit statistics are calculated to assess the goodness of fit between the fitted function and the loaded ("experimental") data. The following parameters are calculated within the boundaries of the fitted region (`Fit from X`, `to X`):

$R\text{-factor} = \dfrac{\sum \lvert y_{\mathrm{exp}}\rvert - \lvert y_{\mathrm{fit}}\rvert}{\sum \lvert y_{\mathrm{exp}}\rvert}$	A perfect fit has an *R*-factor of 0.
$R^2 = \dfrac{\sum (y_{\mathrm{fit}} - \mu_{\mathrm{exp}})^2}{\sum (y_{\mathrm{exp}} - \mu_{\mathrm{exp}})^2}$ with the mean $\mu_{\mathrm{exp}} = \dfrac{1}{n} \cdot \sum y_{\mathrm{exp}}$	A perfect fit has an R^2 parameter of 1.
$\mathrm{SSE} = \sum (y_{\mathrm{exp}} - y_{\mathrm{fit}})^2$	Summed square error (unweighted χ^2). A perfect fit has an SSE of 0.
$\chi^2 = \sum \left(\dfrac{y_{\mathrm{exp}} - y_{\mathrm{fit}}}{\sigma_{\mathrm{exp}}} \right)^2$	If the loaded data does not include weights, χ^2 is automatically computed as unweighted χ^2 (and therefore identical to SSE). A perfect fit has a χ^2 of 0.

7. Non-PCSB Java libraries used in this program

SDAR makes use of the following Java libraries not authored by us:

- Apache Batik SVG toolkit; http://xmlgraphics.apache.org/batik/index.html
- Nelder-Mead Simplex, linear regression and Savitzky-Golay algorithms have been extracted from the libraries by M. Flanagan; http://www.ee.ucl.ac.uk/~mflanaga/java/
- Levenberg-Marquardt in Java by J. P. Lewis; http://scribblethink.org/index.html
- EnabledJComboBoxRenderer by Greg Cope; http://www.algosome.com/articles/enable-disable-jcombobox.html

8. References

1 Weeratunga S, Hu N-J, Simon A & Hofmann A (2012) SDAR: a practical tool for graphical analysis of two-dimensional data. *BMC Bioinformatics* **13**, 201.

2 Hofmann A & Wlodawer A (2002) PCSB--a program collection for structural biology and biophysical chemistry. *Bioinformatics* **18**, 209–210.

9. Copyright, License and Disclaimer