SARI documentation - version noviembre 2025

Alvaro Santamaría

Contents

SARI overview

SARI allows you to visualize discrete time series data, analyze and fit unidimensional models interactively and save the results using a WYSIWYG web interface.

It has been developed in the R programming language, under the interactive framework of the Shiny R package. Currently tested in R version 4.4.3.

The development of SARI started in 2017 to analyze GNSS position time series from the GNSS National Network (RENAG) (RESIF 2017) and it is oriented towards this kind of dataset, but any other type of series can be used as long as the series format is simple, consistent and does not have uncommented alphabetic characters.

SARI is available on all platforms, except mobile devices, and can be accessed in three ways:

- Online from the Shinyapps server
- Offline, from a containerized environment available on DockerHub.
- Offline, after installing the code available on GitHub.

Depending on the way SARI is accessed, there are several ways to run SARI: bookmark on web browser, weblink, Docker Desktop, RStudio, R console or a UNIX-like console. The only requirement is to use a desktop environnment where a web browser can run.

All the different ways to run SARI are implemented in the SARI shell script. This script can be used to start SARI while loading series on a local or remote session. For the ultimate easiness, the SARI shell script also allows creating convenient desktop shorcuts on any system to run SARI with just one click (see an example BAT file here).

This document describes the different options and functionalities implemented in the current version of SARI. Within this document:

* GUI options are given in this `red lettering`.

* GUI specific input values and files are given in *italics*.

* Internal and external links are given in this blue.

Further details and some examples can be found in the SARI official website and in Santamaría-Gómez, A. (2019) SARI: interactive GNSS position time series analysis software. GPS solutions, 23:52. DOI: 10.1007/s10291-019-0846-y

A ~40 min video tutorial is also available on YouTube. This tutorial was made with SARI version "mayo 2021" and does not include the latest additions and corrections, but it is still valid to learn the general usage of the app.

The history of changes and corrections is available in the changelog file.

Current SARI version: *noviembre 2025* – "Our spirit remains in every program we designed for this computer."

Go to top of document

SARI mechanics

The SARI interface is divided into two panels with separate vertical scrolling: the left panel contains all the processing options while the right panel is for visualization of the series and the analysis.
The processing options on the left panel are grouped into five blocks that can be expanded and collapsed with a single click on their header.
Note that some OS and some web browsers apply a display scaling to all web pages by default. It is recommended to adjust the browser zoom (usually ctrl + mouse wheel) so that all the controls, buttons, plots, etc. on both panels fit comfortably on the screen.

I. Input data and format

This block allows uploading and setting the series format, if necessary, before plotting. The input series file must be a text file with standard encoding (ASCII, UTF-8) and organized in columns generally corresponding to the epochs, the observed variables and, optionally, their error bars. The series does not need to be evenly sampled, but only one point per epoch is allowed.
While the following instructions are provided mostly for GNSS position time series, they should also work for any type of time series, provided the file format matches the specifications given below.

There are three different ways to upload a time series on SARI:

1) The user can upload GNSS series from **local files** using the `Input series file` option.
GNSS series are allowed in the *NEU/ENU* format (North/East, East/North, Up) or in the file formats produced by *PBO* (with extension *.pos*, version 1.1.0) and *NGL* (with extension *.tenv3*). For series that are not in *NEU/ENU*, *PBO* or *NGL* format, the user must use the *1D* option, then set the column separator (blanks,

commas or semi-colons) and then set the column numbers containing the epochs, the variable and the error bars if available.

The input file can contain comments anywhere identified by a # at the beginning of the line. These lines will be skipped. In case uncommented text is found, the behavior depends on the requested series format:
Any non-numeric value in a *NEU/ENU* or *1D* series will make the full line to be skipped.
For the *PBO* and *NGL* series, the headers are recognized and skipped, also the first two columns of the *NGL* files and the last column of the *PBO* files. However, any other non-numeric value in these files will stop the app. Data records having a NA, NaN or Inf/inf entries will be treated as valid not-available or not-a-number numeric values and will be automatically skipped, but na, Na, nan, NAN or any other string will be considered as unwanted text.

For *NEU/ENU* and *1D* series, the user may need to set the `Time units` (days, weeks or years) and the `Series units` (metres or millimetres) of the input series, which may be otherwise unknown to SARI (some *NEU/ENU* series are recognized by SARI and the format will be set automatically).
The user must pay special attention to the time units that are set before a *NEU/ENU* or *1D* series are plotted, as these units will define the sampling of the series for the rest of the analysis. Changing the time units after the series have been plotted will change the units of the time axis, but not its sampling. For instance, when changing the time units of an already plotted daily series from years to days or weeks, the sampling units will change from one point per ~0.0027 years into one point per day or per ~0.14 weeks; the already defined sampling period will not change, only its units does. The correct use of several options in SARI depends on defining the correct sampling of the series before plotting them, and this is done by setting the correct time units of the series.
For the *PBO* and *NGL* series, different time units are provided by these formats, so changing the `Time units` is just optional as the sampling period is already known (daily). The latter also applies in case of a *NEU/ENU* series where the epochs are provided in ISO 8601 calendar date and time format.

If the series does not contain error bars, or the user does not want to use them, it is possible to turn the error bars off in the processing using the `use error bars` option. By default, they are always on.

If the format of the input file is not known (content of columns, separation, etc.), it is possible to print on screen the first lines of the input series to assess the corresponding format before plotting using the `show series header` option. After the series has been plotted, if it is not a *1D* series, this option will show the first lines of the coordinate component that has been extracted from the input file and is being used in the analysis.

When the format of the series is set, click the `Plot` button in the Plot controls block.

2) The user can automatically download GNSS series available at specific remote **web servers**.
This is done via the options: `server`, `product` and `station`. To download a series, the user must first select the `server` from those available in the dropdown menu, and then select the `product`. If only one `product` is available for the selected `server`, then it will be selected automatically. With the `server` and `product` selected, SARI will download the list of available stations from the server and for the selected product.
Depending on the connection and the lenght of the list of stations, the download may take up to a few seconds. For some servers for which this operation is sensibly long, a pre-downloaded list of stations is included in SARI allowing for a faster interaction with the remote server. The pre-downloaded lists of stations are updated with each new SARI version.
Once the list of stations from the selected server and product are shown, you can type in the first characters of the `station` name to filter the list of stations available.
When a `station` is selected, the series will be downloaded and plotted automatically, i.e. there is no need to

set the format and units as these are already known.

Similar options are available for downloading a secondary series (see an example below and the Ancillary information block for further details).

3) The values for the `server`, `product` and `station` options can also be provided in any order as a **query string** after the base SARI URL on the address bar of the browser.

These three parameters can be defined for both the primary and secondary series at the same time. For example, adding this query string

?product=SPOTGINS_POS&station=CRAL00FRA&server=FORMATERRE&server2=NGL&product2=IGS14&station2=

at the end of the SARI base URL (both local on RStudio/Docker or remote on Shinyapps) will start a new SARI session, automatically upload and plot the *SPOTGINS* GNSS position series of the station *CRAL00FRA* from the *FORMATERRE* server, and also the *NGL IGS14* series of the station *CRAL* as secondary series (see the Ancillary information block).

This feature allows webmasters of any GNSS series database to include a link on their webpages so that users can open a specific GNSS series directly with SARI on a new browser tab. Contact the author for more information.

At this moment, the following servers and products are available:

Server	Station	Product
LOCAL[1]	file path	NEU/ENU, PBO, NGL, SPOTGINS, 1D
RENAG	4 characters (TLSE)	UGA
FORMATERRE	9 characters (TLSE00FRA)	SPOTGINS, UGA, IGS20, ENS
EPOS	9 characters (TLSE00FRA)	INGV, SGO-EPND, UGA-CNRS, ROB-EUREF
SONEL	9 characters (TLSE00FRA)	ULR7A
IGS	4 characters (TLSE)	IGS20
EUREF	9 characters (TLSE00FRA)	IGB14
NGL[2]	4 characters (TLSE)	IGS20, IGS14, RAPID
JPL	4 characters (TLSE)	REPRO2018A
EARTHSCOPE[3]	4 characters (TLSE)	CWU, PBO, NMT
SIRGAS	4 characters (TLSE)	IGB14
DORIS	4 characters (TLSB)	ESA, GOP, GRG, GSC, IDS, IGN

Server	Station	Product
EOSTLS[4]	14 characters (TLSE_10003M009)	ATMIB, ATMIB(d), ATMMO(o), ECCO(o), ECCO2, ERA5IB, ERA5IB(d), ERA5TUGO, ERA5TUGO(d), ERA5HYD, ERA5HYD(d), ERA5LAND, GRACE, GLDAS2, GLDAS2(d), GLORYS(o), MERRA2ATM, MERRA2ATM(d), MERRA2HYD, MERRA2HYD(d)
PSMSL	Station number	RLR

Notes:

[1] The *LOCAL* server is used only to upload series from local files on a local SARI session (more details in the SARI shell script).

[2] The *RAPID NGL* series correspond to the IGS14 solution.

[3] The EarthScope (UNAVCO) series correspond to the *IGS14* solution.

[4] The DORIS series correspond to the *25wd01* solution.

[5] The *EOSTLS* loading series are those computed in the center of figure (CF) frame. WARNING: some of the *EOSTLS* series are very long and/or have a very high sampling, loading these series may take longer than expected. It is recommended to use the daily series from those products marked with *(d)*, if they are available. Products marked with *(o)* represent old products that are no longer updated.

Contact the author if more servers or products need to be added.

When uploading a series from a remote server via the the `server`, `product` and `station` options, SARI will also upload the corresponding *sitelog*, *soln.snx* or *custom offset* files, depending on the selected server, if these files are available for the selected station (see the Ancillary information block).

Once uploaded and plotted (see the Plot controls block), the series will be shown on the visualization panel on the right. If the series contains three coordinate components (all except the *1D* format), the three coordinate components of the series will be shown together in the *3D* tab. Each coordinate component will be also available individually in separate tabs on the top. The series analysis is done separately for each coordinate component in its respective tab (see the Fit controls block).

In case the coordinate components are not known to SARI, the tabs will be labeled : 1st, 2nd and 3rd component, respectively.

When plotting a primary and a secondary series with different formats, the option `N@E` can be activated to match the coordinate components North and East of the secondary series with respect to those of the primary series (see the Ancillary information block).

For GNSS time series, the station ID is extracted from the first characters of the input file name and will be shown in the `series ID` box under the loaded series. The station ID can be modified by the user if necessary. For more information on the use of the station ID, see the Ancillary information block. For other type of series, this feature can be neglected.

It is possible to average and reduce the sampling of the input series, for instance from daily values to weekly, by using the `reduce sampling` option. The user needs to provide the new sampling period in the same time units of the series. This option also accepts mathematical expressions identified by a starting =, such as *=7/365.2425* for reducing a daily sampling into weekly sampling using (Gregorian) year time units.

Note that, when the sampling of the series is daily or weekly (as is common for GNSS position time series), it is preferable to change the time units to days or weeks, reduce the sampling of the series using an exact period given in days or weeks, and then change the time units back to years if necessary. This way, the new resampled epochs will be more precise than if an approximate value for the number of days in a year is used (i.e. *365.2425* for the Gregorian year used by SARI).

If high-frequency data are not needed, averaging the series will save a lot of time when fitting models, finding discontinuities or when estimating the colored noise in the series (see the Additional fit block).

If the series was computed with a constant integration period, like the common 24 h in GNSS position series, this option is useful to transform daily GNSS series with irregular sampling into daily GNSS series with regular sampling, i.e., points separated by exactly 24 h, except when data gaps exist.

If there were points already removed from the series before reducing the sampling, these points will not contribute to the new averaged values. Also, if there is a secondary series being used with the `correct` or `average` options, the reduced sampling will be obtained before the secondary series is used. This means that the primary and secondary series would be compared after their original sampling has been reduced.

A summary of the contents of the plotted series (number of points, time range, sampling, etc.) will be shown at the bottom of this block.

If the station coordinates are already known, or provided by the user, the `show location map` option will show a map with the station's location together with the Bird (2003) plate boundaries. The station ID and the plate name are shown by hoovering over the map.

Go to top of document

II. Plot controls

This block allows working with the plots of the time series.

Once the format of the series is set, they can be plotted or resetted with the `plot` and `reset` buttons, respectively. The `reset` button will also remove the uploaded series and any other information given by the user (see the Interactive operation section).

The button `overview` opens a new browser window containing a static PNG figure with the series of the three coordinate components taking into account the current state of the series (zoom, removed points, plate motion correction and fitted model). If the series have been fitted, this figure will also include the estimated linear trend and the standard deviation of the model residuals for each fitted component.
There is no need to close and reopen the `overview` window, if the windown is left open (for instance, on a second screen), the PNG figure will update automatically each time there is a change in the series or in their fitting.

The `truncate` option allows removing the beginning and/or end of the series up to/from a given time epoch. The epochs must be given in the same time units as those selected for the series (see the Input data and format section).

The `remove period` option allows removing all the points inside the given *from* and *to* epochs. The epochs must be given in the same time units of the series.

Outliers selected manually can be excluded from the analysis with the `toggle` option. Outliers can also be excluded automatically with the `auto toggled` button after providing a residual threshold or a normalized residual threshold (i.e., times above the error bar, if provided).
The `reset toggle` button restores at once all the points that were previously toggled.
See more details about removing/restoring points in the Interactive operation section.

The `all components` option changes between removing outliers for each coordinate component independently or

from all components simultaneously. This option works also when analyzing different columns using the 1D series format.

The `permanent` option will permanently delete the next points to be toggled/truncated from the series. Permanently means the selected points cannot be restored back and they will not be shown again in the current session. This option is intended, for instance, to remove extreme outliers that do not allow visualizing the series correctly.
The original series file is not modified and can be reloaded to use all the points again, after resetting the current session with the `reset` button. The `permanent` option is deactivated by default, and once one or more points are toggled/truncated in a single action, it will automatically deactivate itself so the next points to be toggled/truncated will be available to be restored. When toggling/truncating points that were already removed, this option will delete them permanently as well, instead of restoring them.

The `include in file` option will keep the excluded outliers in the downloaded results file as commented lines. No fitting values will be provided for these points.

The `scrolling` option enables/disables the vertical scrolling of the left panel. By default the scrolling is enabled. When disabling it, the user will be able to take a screenshot of the full web page with the plots and all the input parameters that were used to make the plots, which is very convenient for sharing and archiving a specific analysis, making reports or marking assignments. The quality of the full page screenshot may depend on the browser/extension used.

Alternatively, the user can download the full SARI interface from the web browser (usually by pressing Ctrl+s) into an HTML file together with the corresponding web files in a separate directory having a similar name. The downloaded HTML file represents a frozen SARI session (i.e., not connected to the server) that can be reopened later on the web browser even offline. The advantage of the HTML page compared to the fixed screenshot is that all the numeric values are available to be selected and copied, so the same analysis can be easily replicated in a live session. However, before opening the HTML file in a web browser, it needs to be modified first. The shell script SARIwebpage.sh does the modifications automatically.
Note: Unfortunately, at this moment, the values of the UI interface, including the options selected by the user, are only saved in the HTML file when using Mozilla Firefox. Other web browsers, like Safari and the Google Chromium family (Google Chrome, Microsoft Edge and Opera), do not keep this information when saving the web page.

Go to top of document

III. Ancillary information

This block allows for the uploading of files containing complementary information related to the analysis of GNSS position time series. When an ancillary file is uploaded, the file contents can be visualized in a new browser tab by clicking on the link at the right of each input file.
The user can upload any combination of the following possibilities:

- a file with the LS or KF model parameters from a previously saved SARI file
- a GNSS *sitelog* file (an example of this file can be found here)
- a GAMIT-like *station.info* file (an example of this file can be found here)
- an IGS-like soln discontinuity file (an example of this file can be found here)
- a customized offset file to plot specific dates (for instance, earthquakes)
- a secondary series that can be `shown` next to the primary series in green color for comparison purposes and detection of common features, or used to either `correct` the primary series to analyze the difference or `average` both the primary and secondary series values. For instance, a model can be subtracted a posteriori from the series (loading, post-seismic, etc.) or even the series of a nearby station in order to remove the

spatially correlated signal/noise and better detect the individual equipment changes, relative motion or common variations (see the notes below).

- a plate motion model to be `shown` on top of the horizontal component series or `removed` from the series. The user needs to select a plate motion model from the dropdown menu and enter the name of a tectonic plate, or provide a custom plate motion model for the current series, or provide the parameters of the Euler's pole manually (see the notes below).

- a vertical land motion prediction from a GIA model to be `shown` on top of the vertical component series or `removed` from the series (see the notes below).

Notes on the equipment change logs:

1. The *sitelog*, *station.info*, *soln* and *customized* files can be uploaded to list/plot equipment changes and help identify position discontinuities. Antenna changes are represented by solid lines. Receiver and other changes represented by dashed lines. Each type of file has a different line color: *sitelog* in blue, *station.info* in pink, *soln* in gray, *customized* in cyan.

2. The dates from the *sitelog*, *station.info* and *soln* files will be transformed according to the time unit selected for the series. If the time unit is days, the dates will be transformed into Modified Julian Days. If the time unit is weeks, the dates will be transformed into GPS weeks.

3. The station ID (see the Input data and format block) will be used to extract the information from the *station.info*, *soln*, *customized offset*, and *plate motion* files. The user is responsible for checking that the station(s) in these files actually correspond to the same station ID being used (many GNSS stations around the world share the same ID).

4. The *customized offset* file must contain one or two columns separated by spaces or tabulations. The first column contains the date in the same time units as the input series. The second column is optional and contains the station ID. If the second column is present, only the dates corresponding to the current station ID will be extracted. If the second column is missing, all the dates in the file will be used. A warning will be shown if the file contains more than two columns.

5. As an exception, the discontinuities file from the Geodesy Plotter and from the Nevada Geodetic Laboratory can be uploaded directly as a *customized offset* file without changing their format.

Notes on the secondary series:

1. The *secondary series* can be obtained from a single file or from several files uploaded at the same time. When uploading several files, they will be added together to form a single *secondary series*. For instance, atmospheric, oceanic and hydrological loading series at the same site can be uploaded at the same time to obtain a single *secondary series* representing the total loading at the site.

2. When loading several files, only the common epochs among them will be retained. If one epoch is missing in one of the files, it will be skipped.

3. Several loading products can be selected and uploaded directly from the EOSTSL server (see the Input data and format block). If necessary, the series are automatically resampled to daily before being added together to form the *secondary series*. Change the averaging period of the *secondary series* to obtain a different sampling (e.g., weekly or monthly). The recommended loading models (J-P. Boy, personal communication) are: atmospheric

loading from the latest ECMWF ERA5 reanalysis including the dynamic ocean response from the TUGO-m barotropic model (ERA5TUGO(d) label) together with hydrology loading (soil moisture and snow) from the ERA5-Land reanalysis (ERA5LAND label). More details at the EOSTLS server.

4. If several *secondary series* are uploaded manually by the user, the series are added together as they are provided, so it is the user's reponsability to verify that they have consistent time units and sampling.

5. The `Swap` button exchanges the role of the *primary* and *secondary series* so that the *secondary series* can be now analysed as if it was loaded as the *primary series*.

6. When uploading a *secondary series*, both station IDs will be shown in the `series ID` box on the left panel, together with a "&" when using the option `show`, a "−" when using the option `correct`, a ":" when using the option `average`, or a "+" when using the option `merge`. When the *primary series* is combined with a *secondary series* (using `correct`, `average`, or `merge`, but not `show`), the IDs from both series will be extracted from the *station.info*, *soln*, and *customized offset* files. Otherwise only the ID from the primary series will be used.

7. If the `show` option is selected, the *secondary series* will be plotted in green on the right y-axis and will not be included in the processing. This means that the *secondary series* could have a different sampling than the *primary series*. In case the `correct` or the `average` option is selected, only the common epochs between the *primary* and the *secondary series* will be shown. In the latter case, the *secondary series* must have observations at common epochs with the *primary series* because the series are neither filled nor interpolated at common epochs. For the `merge` option, only the epochs that are present in the *secondary series* and not in the *primary series*, will be added to the *primary series*.

8. When the *primary series* is combined with a *secondary series* using any of the `correct`, `average`, or `merge` options, both the `plate model` and `GIA model` options will be disabled and fixed to their current status. This is done to avoid using these model corrections before and also after the *primary* and *secondary series* were combined. The `plate model` and `GIA model` options must be used before combining the *primary* and *secondary* series.

9. For typical GNSS series with daily sampling, even if the sampling period is constant, very often the epochs will not exactly match between the *primary* and *secondary series*, especially if the series are produced by different people. If the *primary* and *secondary* series have both a constant sampling of one day, but their epochs do not match, SARI will try to compute the constant fraction of a day shift between both series and apply it to the *secondary series* to match the epochs of the *primary series*. This feature will only work correctly if both the *primary* and *secondary* series have a regular sampling, independently of their time unit.

10. If the *secondary series* does not have the same sampling as the *primary series*, a warning will be shown on screen. The sampling of the *secondary series* can be reduced with the `averaging` option, which is equivalent to the `reduce sampling` option of the *primary series* (see the Input data and format block).

11. For NEU/ENU series, the *secondary series* must have the same format than the *primary series*. For 1D series, the user has the option to choose the column number and column separator for the *secondary series* independently of those already set for the *primary series*.

12. The units of the *secondary series* can be scaled using the `scale factor` option. Also the y-axis values of the

secondary series are partially controlled by the `same scale` and `same axis` options with respect to the left y-axis of the *primary series*. The `same scale` option will force the right y-axis to have the same range as the left y-axis, but not necessarily the same values. The `same axis` option will force the right y-axis to be the same as the left y-axis.

13. The `full series` option will expand the plot area to show both the *primary* and *secondary series*. By default, the plot are only includes the *primary series*.

14. If the coordinate components of either the *primary* or the *secondary* series are unknown, the `N@E` option can be used to swap the North and East components of the *secondary series* to match the components of the *primary series*.

Notes on the plate motion model:

1. In order to compute the *plate motion*, SARI needs to know the station coordinates and the parameters of the Euler's pole. The station coordinates will be extracted automatically from *NGL*, *PBO* and some *ENU/NEU* series. For other series formats, the Cartesian (X,Y,Z) or geographic (latitude, longitude) coordinates of the station must be provided below the `select a plate model` option in this section. The units of the Cartesian station coordinates must be the same as the time series units (i.e., meters or millimeters). If geographic coordinates are provided and the `Series units` have not been set, SARI will try to guess if the values of the GNSS series are in meters or millimeters and then transform the geographic coordinates into Cartesian coordinates with the correct units. This guess may fail if the series are ambiguous, so either the series units must be set by the user or the station Cartesian coordinates should be preferred as input.

2. The `Select a plate model` option allows selecting any of the three *plate motion* models implemented in SARI: the ITRF2020 model (Altamimi et al. 2023), the NNR-MORVEL56 model (Argus et al. 2011) and the NNR-GSRM v2.1 model (Kreemer et al. 2014). The parameters of the Euler's pole for each plate can be shown by clicking on the `Show the selected plate model` link. The user needs to select a plate motion model and then select one of the tectonic plates available for each model.

3. Alternatively, the user can `Upload a custom plate model` that must have six or seven values per line corresponding to the station ID (first value), the station coordinates (two values: latitude & longitude; or three values: X,Y,Z), and the Euler's pole parameters (three values: latitude,longitude,rotation or X,Y,Z rotations). If a station ID is present in more than one line, only the last one will be used. The units of the plate model parameters are degrees for latitude & longitude and degrees/Ma for rotations. An example of the custom *plate motion* model can be found here).

4. Once the *plate motion* model is read, the extracted values will be shown on the left panel. The user can check and modify these values manually. If no values are being shown, it means no values were found for either the selected plate (see point 2) or the current station ID (see point 3) that is shown in the `station ID` box of the left panel.

5. The `plate model` option will be disabled and fixed to their current status (*none*, *show* or *remove*) if both the *primary* and *secondary series* are combined using any of the `correct`, `average`, or `merge` options. This is done to avoid using the `plate model` correction before and also after the *primary* and *secondary series* were combined. The `plate model` option must be used before combining the *primary* and *secondary* series.

Notes on the GIA model:

1. The vertical land motion (VLM) linear trend value is extracted by bilinear interpolation of the station's location (also for the secondary series) on the selected model grid.
2. The available GIA VLM grids are obtained from the Caron and Ivins, 2019 model with resolution 1°x1° (using the total mass sources), the ICE-6G-VM5a model with resolution 0.2°x0.2° (version D, Peltier et al., 2018) and the ICE-6G-ANU model with resolution 1°x1° (Purcell et al., 2016).
3. The station coordinates must be provided in the plate motion model section (see the notes above).
4. The extracted VLM value can be modified manually to change the sign, for instance if correcting the VLM of a TG series, or to use a different VLM value, for instance obtained from a GNSS series.

5. The `GIA model` option will be disabled and fixed to their current status (*none*, *show* or *remove*) if both the *primary* and *secondary series* are combined using any of the `correct`, `average`, or `merge` options. This is done to avoid using the `GIA model` correction before and also after the *primary* and *secondary series* were combined. The `GIA model` option must be used before combining the *primary* and *secondary* series.

Go to top of document

IV. Fit controls

This block allows fitting a model to each one of the coordinate component series using

- Weighted least squares (LS)
- First-order extended Kalman filter/smoother (EKF)
- Unscented Kalman filter/smoother (UKF)

The `Select model components` option allows defining the fitted deterministic model by including any combination of linear trend, higher-degree polynomial, offsets/discontinuities, sinusoidal signals, exponential and logarithmic decays.

Some of these components require additional parameters to be included in the model (e.g., reference epochs, periods, a priori values, etc.). The reference epochs and the a priori values can be left empty and SARI will use some reasonable values automatically. Note that the a priori values can have a strong impact on the quality of non-linear fits, for example the exponential and logarithmic decays (see the notes below).
Additionally, if using a *KF* fit, the initial state and the standard deviation of the process noise can be set for the trend and sinusoidal components (see the notes below).

If using a *LS* fit, the `segmented model` option allows fitting the linear rate, sinusoidal and polynomial components to different segments of the series, which are defined by a comma-separated list of *break epochs*. If the input list of *break epochs* is not empty, a new option allows fitting a *continuous* or *segmented* linar rate, sinusoidal variation or higher-degree polynomial. In case the *continuous* option is selected, the break epochs will be ignored for a specific component. For instance, a continuous sinusoidal variation can be fitted together with a segmented linear rate, or vice versa. In case the *segmented* option is selected, the estimated model components will not be constrained between the different segments.

The fitted model will be represented by a red line on top of the original series and in the plot of the residuals with a zero value line.
The `click and collect` option allows adding the epochs of the clicked points on the series plots automatically to the list of offset epochs in the fitted model.
The fitted offsets will be represented by vertical red lines in the residual plot.

A fit summary will be shown on the visualization pannel on the right.

In case of a *LS* fit, the summary includes the estimated values of the chosen model components, their formal uncertainty, their absolute Student's t-test statistic and their corresponding significance level, i.e., the probability of rejecting a true null hypothesis in which the parameter value is not significantly different from zero; in short, the probability that the model parameter is unnecessary. Probabilities higher than 5 % will be printed in red as a warning. The correlation between the model parameters is also provided on the visualization panel. Similarly, correlations higher than 0.8 will be printed in red as a warning.

The LS fit also prints the estimated "residual standard error" value, which, in this context, represents the unitless scale factor that was applied to the series error bars to obtain the residual error bars. Values higher/lower than 1 ususally indicate that the series error bars are too optimistic/pesimistic to represent the unexplained variability of the series themselves.

In case of a *KF* fit, the summary will include the statistics (mean value, standard deviation, etc.) for each model component included in the estimated state.

When the three coordinate components are fitted (excludig 1D series), the tab *residuals* will appear at the top. This tab will show the plots of the three coordinate components, as in the 3D tab, but using the fit residual series. By comparing the three residual series at the same time, it should be easier to verify the general quality of the fit. The `save` button will download the three residual series into a single file.

The `search discontinuities` button provides an automatic guesstimate of the location of probable discontinuities in the time series. The algorithm is based on the assessment of deviations from stability in different segments of the series using a classical linear regression model and a Bayesian Information Criterion (BIC). The algorithm is described by Zeileis et al. (2003). WARNING: this option is very time-consuming (see the notes below).

Notes on the Kalman filter:

1. The EKF/UKF fits include both the forward Kalman filter and the backward Kalman smoother using the Rauch-Tung-Striebel (RTS) algorithm. Only the smoothed solution is kept.

2. The UKF implementation is described by Julier and Uhlmann (2004) and the unscented RTS smoother is described by Särkkä (2008).

3. The KF fit does not include any dynamics (i.e., control inputs) and assumes additive, independent measurement and process noises; the latter for the trend and the sinusoids only. The measurement noise is time invariant, unless the series error bars are in use. In that case, the relative change of the error bars is used to scale the measurement noise. The average of the actual measurement noise level used for the entire series will correspond to the measurement noise value indicated on the left panel. Data gaps are also accounted for by increasing the process noise through the missing observations.

4. The user can set the expected (average) measurement noise level and the initial values and uncertainties of the state for all the parameters being fitted or leave them blank. In the latter case, some values will be proposed. Note that the filter output strongly depends on all the provided initial values. For instance, an extremely small initial uncertainty on a parameter would "fix" it to its initial value; which is useful when we know precisely the magnitude of a discontinuity, for instance.

5. The measurement and process noise levels need to be provided with respect to the series sampling that is being used in the fit, which is indicated on the left panel at the bottom of the Input data and format block.

6. The `compute measurement noise` option will provide the most likely measurement noise level, within some bounds provided by the user, and with respect to the process noise also provided by the user. This is done through a maximum likelihood optimization of a preliminary UKF fit with respect to the provided process noises for the trend and sinusoids only. The optimization uses the Brent (1971) method, and it may take up to ~20 iterations of a standard UKF fit. The 95 % confidence interval of the estimated measurement noise standard deviation will be shown next to the input bounds on the left panel.

7. The process noise values for the trend and sinusoidal variations are zero by defaul. Any positive value will make these parameters to change with time. When fitting a time-variable trend, a plot below the residual series will show the time series of the estimated time-variable trend, which is useful to assess whether the filter is too smooth or not, especially if other parameters were fitted together with the rate.

8. When fitting a time-variable sinusoid, there is an option to apply the process noise to the sine component (S) only or to both the sine and cosine (C) components independently. The first option will capture amplitude variations, but also phase and maybe even frequency variations. The second option may only capture amplitude variations, but this is not guaranteed as the estimates of the sine and cosine terms are not constrained to vary the same amount.

9. Due to the KF fit being slower than the LS fit (see details in the Known issues section), the user has to set all the parameters of the filter first and then push the `run KF` button. This way, the filter is run once and not each time one of the parameters is changed, as for the LS fit. The parameters left blank will be completed automatically (at the user's risk).

10. If the user has fitted the same model with LS before fitting it with the KF, some of the LS-estimated values will be used as the initial state for the KF.

Notes on the sinusoidal fitting:

1. By default, the reference epoch of the sinusoidal fitting is the mean of the available measurement epochs, and phase values are estimated accordingly.

2. The sinusoidal model is linearized by separating it into sine and cosine components. The estimated sinusoidal amplitude and phase, including their formal uncertainty, are provided below the correlations of the fitted parameters in the fit summary of the right panel.

3. The convention for the sinusoidal phase is that the sine component is positive clockwise.
4. Below the option to enter the `sinusoidal periods`, the `draconitic harmonics` option automatically adds 4 overtones of the draconitic orbital period for each selected GNSS constellation. Choose the constellation(s) that was used to compute the position time series. The number of draconitic harmonics can be modified in the `sinusoidal periods` input.

Notes on the exponential/logarithmic decay fitting:

1. The expected exponential/logarithmic decay fit corresponds to that commonly observed in some GNSS position time series due to visco-elastic deformation from post-seismic or surface mass loading phenomena. The decay is assumed to be asymptotic towards a more recent date and lasts for several years. The internal algorithm computes its own a priori values based on this assumption, but the user is free to provide their own a priori

values when fitting a different decay.

2. The success in fitting an exponential/logarithmic decay depends heavily on the quality of the a priori values for the asymptotic offset and the decay rate parameters. If the user does not provide these a priori values, a built-in algorithm provides a heuristic guess. The quality of this guess depends on the amount of data available shortly after the start of the decay and whether the series contains other components that may alter the decay, such as discontinuities, trends, periodic variations, or noise. Data gaps can be particularly problematic for accurately guessing the a priori values. If a data gap occurs at the beginning of the decay, masking the epoch of start, it is recommended to set the reference epoch of the decay right before the first available observation and estimate a position offset at this epoch.

3. For each decay being fitted in the series, if any of the a priori values for the asymptotic offset and the decay rate is empty or deleted by the user, the built-in algorithm will kick off and update both values with its own guess.

4. If more than one model component is occurring simultaneously (e.g., a position offset with an exponential decay or an exponential decay with a logarithmic decay), the quality of the guessed a priori parameter values may be poor because they are obtained independently for each decay.

5. If the a priori parameter values are not accurate enough, the fitting may provide unsatisfactory results or even fail. In this case, the a priori parameter values can be manually tuned on the left panel using an educated or an eyeball guess. For instance, the exponential/logarithmic asymptotic offset corresponds to the observed difference between the series value at the start of the decay and the asymptotic value (the asymptotic offset is positive if the series is decreasing for the exponential and increasing for the logarithmic). If the decay complies with the expected behavior in GNSS (see point 1 of these notes), the decay rate should be positive. For the exponential, it is roughly the time taken to reduce the asymptotic offset to one-third of its value; for the logarithmic, the decay rate can be challenging to guess accurately.

6. If more than one decay (exponential, logarithmic or mixed) is being fitted, and the a priori values for one of them are changed (improved), it may cause the fitting to fail. This is because the a priori values left unchanged for the other decay may now be worse than before, i.e., with respect to what remains to be fitted.

7. Finally, bear in mind that even if the epoch of an earthquake is precisely known, down to the minute or even the second, a GNSS time series is a discrete series computed very often out of daily batches. This means the discontinuity may appear in the GNSS series at the daily batch boundaries and not at the earthquake epoch itself. If the discontinuity epoch is not correctly set with respect to the discrete series, i.e. at the epoch of the observed discontinuity, that may be slightly different than the one from the earthquake, the fitting of the post-seismic relaxation may provide bad results or even fail. Excluding the GNSS estimate of the day of the earthquake may be enough to fix this problem.

Notes on the discontinuity detection:

1. The success of the automatic detector improves when any systematic variation in the series is removed beforehand, such as trends, periodics, decays, outliers. Therefore, the detector can only be applied to residual series. The user can even remove some discontinuities before running the detector or remove & run it several times as it may detect new discontinuities when the residual series flattens.

2. The automatic detector has difficulties in detecting discontinuities when approaching the beginning and the end of the series. To minimize this limitation, the series is automatically padded with random white noise at both ends (only for the automatic detection run). However, this may sometimes provide false positives extremely close to the beginning or end of the series due to an unwanted change in the local mean value within the padded values. The user should be suspicious of these positives unless they are significant and really evident.

3. The user can set the minimum length of the segments considered in the stability test, which is 10% of the series length by default. However, it must be large enough to include at least three observations. The length of the segments will impact the number of discontinuities found and the processing time, which increases inversely proportional to the segment length. The processing time also depends on the number of observations. Therefore, do not hesitate to `reduce the sampling` as necessary (see the Input data and format block). Additionally, discontinuities are often easier to find in downsampled (filtered) series.

4. The epochs of the detected discontinuities (if any) are displayed on the left panel. The user can copy and paste the epochs into the model components to remove them. Note that in case the series are auto-correlated, the detected discontinuities could be caused by random non-zero mean noise fluctuations (see the notes on the offset verification below).

5. Using 75 synthetic series from the RENAG "DOGEX" experiment (from Stephane Mazzotti), the automatic detector performed as good as my own eyes for low noise series (horizontal components). However, manual detection was still much better for noisy series (vertical component).

Notes on the offset verification:

1. Spurious discontinuities of unknown origin (i.e., no recorded GNSS equipment changes or earthquakes) may appear in series containing time-correlated noise. The corresponding estimated offsets may be considered statistically significant if the series is assumed uncorrelated. This happens in the implemented LS fitting and when using the *automatic discontinuity detection.*

2. To cope with this situation, the user has the option to verify the significance of the estimated offsets against time correlation in the series provided by a power-law noise process, which is the most common in GNSS position series.

3. The parameters of the power-law noise process must be provided from a priori knowledge or estimated with the `Noise analysis` option (see the Additional fit block). In case a noise analysis was already run, the estimated noise values will be included automatically in the offset verification section.

4. In the offset verification run, a Generalized Likelihood Ratio test is used to compare the (log-)likelihood of the residual series (offsets removed) being produced by the provided noise model compared to the (log-)likelihood of the series with each one of the estimated offsets restored back individually. The higher the likelihood ratio, the greater the chances a particular offset is likely not being produced by random noise fluctuations.

5. The significance of each estimated offset is provided on the left panel based on the probability of a centred chi-squared distribution. Usually, only offsets with a probability higher than 95 % should be retained in the fitted model. Otherwise, there are chances that the estimated offsets may be generated by random noise instead of being systematic changes.

Go to top of document

V. Additional fit

This block allows for additional time series fitting, including:

- An automatic linear trend estimator using the Median Interannual Difference Adjusted for Skewness (MIDAS) algorithm (Blewitt et al. 2016). Depending on the linearity of the time series, this algorithm provides reasonable and robust linear trend estimates in the presence of position discontinuities in the series (see the notes below).

- A pseudo-automatic linear trend estimator using the minimum entropy method. This option provides reasonable linear trend estimates in the presence of near-stationary unknown colored noise, if the epochs of the discontinuities in the series are provided (see the notes below).

- The histogram of the original, model, residual or smoothed series with its expected normal distribution (red curve) and a stationarity assessment using the Augmented Dickey-Fuller and the Kwiatkowski-Phillips-Schmidt-Shin tests.

- A non-parametric waveform for studying periodic patterns in the residual series not having a sinusoidal shape. The waveform is plotted in red as if it were part of the fitted model (see the notes below).

- The amplitude or power spectrum from the weighted Lomb-Scargle periodogram of the original data, the fitted model, the model residuals, the smoothed values, or the smoother residuals. These five spectra can be plotted and compared against each other (see the notes below).

- A pseudo discrete wavelet transform analysis for irregularly sampled time series described in Keitt (2008) and available for the original data, the fitted model, the model residuals, the smoothed values, or the smoother residuals. WARNING: this option is very time-consuming (see the notes below).

- The Vondrák (1977) band-pass smoother to reduce the variability around chosen periods from uneven, unfilled, uninterpolated, and uncertain sampled observations or evenly sampled observations with gaps and varying error bars. The smoother can be applied either to the original series or to the residual series from the model fit. It is plotted in yellow in both the series and residual plots (see the notes below).

- A noise analysis to estimate the full variance-covariance matrix that best describes the model/filter residuals as a Gaussian process by means of an MLE of the parameters of a chosen covariance model. WARNING: this option can be very time-consuming (see the notes below).

Notes on the MIDAS trend estimates:

1. The algorithm was implemented based on the description by Blewitt et al. (2016) and using Geoff Blewitt's code available here.

2. Only data pairs strictly separated by one year (within a tolerance of half a day) are used in the computation.

3. Contrary to the original algorithm, the estimated velocity uncertainties are not scaled.

4. Outliers removed from the series will not be considered in the selected data pairs, which is useful when having periodic outliers, for instance due to snow/ice on the GNSS antenna.

16

5. If the user removes discontinuities from the series, two MIDAS estimates will be provided: with and without the interannual differences crossing the discontinuities. The latter is the one that should be considered. Both estimates are provided so the user can assess the level of robustness of the trend estimate when the algorithm is applied blindly with series having known discontinuities, but not removed, or having unknown discontinuities, i.e., not recorded in any discontinuity database.

6. The histogram of the selected interannual trends (discontinuities skipped) is also displayed to provide a qualitative assessment of the MIDAS estimate.

Notes on the minimum entropy trend estimates:

1. The method provides the linear trend that minimizes the Shannon entropy of the detrended series, i.e., the linear trend that maximizes the predictability of the series. The entropy is approximated by the non-parametric Vasicek estimator based on sample-spacings defined by a window size.

2. The implemented algorithm is based on the one proposed by Saleh et al. (2024). The main deviations of the algorithm include an adaptive window size and the estimated velocity uncertainty that accounts for the number and location of the discontinuities.

3. At this moment and because of position discontinuities, the algorithm samples a wide range of possible velocity values with a reasonably high resolution and provides the one with the highest score. In the future, a more efficient algorithm could make use of an optimization method to find the solution faster by minimizing the amount of sampled values.

4. The algorithm assumes long, near-stationary and iid residual series, which is rather difficult to satisfy with typical GNSS position series. Nevertheless, it seems to provide reasonable linear trend estimates with realistic uncertainties in the presence of unknown colored noise, as long as the noise remains near-stationary, which may be roughly assessed using the `histogram` and `periodogram` options.

5. The algorithm is robust against a few outliers and short-period oscillations. However, the epochs of the discontinuities, if any, must be indicated by the user. The algorithm estimates the weighted sum of the entropy for each segment between discontinuities.

6. If there are discontinuities already used in a LS fit, they will be automatically included in the entropy estimate. If there is no LS fit or there are missing discontinuities in the LS fit, the user can indicate the epochs of the discontinuities under the `entropy` option on the left panel.

7. The discontinuity epochs provided for the entropy estimate will not be used in the LS fit, but any discontinuity in the LS fit will be used in the entropy estimate.

Notes on the waveform:

1. The non-parametric waveform is estimated by averaging the model or smoother residual series on coincident epochs within the period defined by the user. The series must be stationary, so this feature can only be applied to residual series (assuming they are near stationarity). For unevenly sampled series, the waveform is approximated by averaging almost coincident epochs into small bins around typical epochs. This is neccesary

when the time unit is years and the series sampling is in days, as commonly found in GNSS series. In this case, non-coincident observation epochs are unavoidable due to the fact that the Earth's orbital period is not a multiple of the Earth's rotation period. Note that epochs or epoch bins containing few averaged values may provide a biased value of the waveform for that epoch, for instance with extremely uneven sampling.

2. Contrary to the sinusoidal fit, the non-parametric waveform can take any shape. On the other hand, it will not be defined by an analytical expression. Neverthelesss, the waveform can still be used to predict unobserved series like data gaps or past/future observations.

3. By default, the estimated waveform will be added to the fitted model, or smoothed series, and will be subtracted from the residual series for visualization purposes, without actually modifying the original series when saving the results. The estimated waveform series will be saved in a separated column of the downloaded file.

4. The waveform is estimated from the residual series and, by default, it will not affect the fitted model parameters unless the `remove from series` option is activated. In this case, the estimated waveform will be extracted from the original series before the model fit, affecting the estimated model parameters. This modification of the original series can be undone by deactivating the `remove from series` option. Note that any modification of the model components or the original/residual series (i.e., outliers) will imply deactivating the `remove from series` option automatically.

Notes on the periodogram:

1. The amplitude spectrum is useful for quickly assessing the amplitude of the different sinusoidal peaks that may compose the series. The power spectrum is similar, but in squared units, and it is useful for quickly assessing the auto-correlation of the series.

2. The *oversampling period* and both the *max/min periods* to be estimated can be set by the user. The period units are the same as the series time units. The frequency resolution of the periodogram is set as the inverse of the maximum period divided by the oversampling value. By default, there is no oversampling applied, i.e. unity value. The larger the number of periods to estimate, the longer it will take to estimate the periodogram. Generally, it should be computed reasonably fast unless the user asked for a very large number of periods, for instance, from long series with high sampling.

3. With no input from the user, the *maximum period* and a reasonable *minimum period* will be provided. The minimum period is based on the assumption that observations are already integrated over a constant observing period given by the most frequent time spacing in the series. This pseudo-Nyquist period should be fine for regular sampling with gaps, as in a typical GNSS position series.

4. In case the series are made up of instantaneous observations (or that could be considered instantaneous) with uneven sampling, rather than being integrated over an observing period, the actual Nyquist period exists even for extremely uneven series and can be much shorter than the typical/average sampling of the series, as proposed by Eyer and Bartholdi (1999). It is the user's responsibility to verify and set the minimum period that would allow recovering the maximum of frequency information from the series.

5. The user can zoom in on an area of the periodogram (see details in the Interactive operation section) and increase the *oversampling* to get a better resolution of the amplitude/power distribution around specific periods.

Note that "better resolution" could be misleading here as there is no new (i.e., independent) information being included in the periodogram when the *oversampling* is increased; the periodogram looks just smoother. If a zoom was applied, the periodogram will be recomputed with the new oversampling for the selected area only, which saves processing time by reducing the number of periods to compute. The *maximum* and *minimum* periods of the new periodogram will change on the left panel. To compute the full periodogram again, the user must delete one of the *maximum* or *minimum* periods, and the full range will be used again. It is recommended to set the *oversampling* back to 1 (default) before computing the full periodogram again.

6. The implemented algorithm forces the integral of the power spectrum to equal the total variance of the observed series. That is, this is a different normalization compared to the standard Lomb-Scargle periodogram. This allows for a more intuitive and direct comparison of the power spectrum of the different series (e.g., original vs filtered).

7. In both the amplitude and power spectra, if the user clicks on the periodogram, the coordinates (period & amplitude/power) of the clicked point will be shown under the plot together with those of the closest estimated period; the latter point will be marked on the periodogram plot. In addition, if no oversampling is applied (unity value), the scatter (standard deviation) of the series integrated up to the clicked period will also be shown under the plot. For instance, if the user clicks at the longest period of the series or beyond, and the computed periodogram includes all possible periods, the obtained value will nearly correspond to the standard deviation of the series.

8. In case the power spectrum is selected, two lines will pe plotted on top of the periodogram: a pink dashed line representing the slope of a pure flicker noise process commonly seen in residual GNSS series, and a solid line with the same color as the periodogram itself representing the average slope of the rightmost type of periodogram selected amongst the five options (original, model, residuals, etc.). These lines should help the user to roughly assess the level of time correlation of the series. However, if the user runs a `Noise analysis`, the power spectrum of the estimated noise model will be plotted instead of the aforementioned lines. This will help the user to assess the quality of the estimated noise model and to assess the crossover period between the components of the noise model (if more than one).

9. If the user wants to use the `Noise analysis` option to fit a power-law noise model, it is highly recommended to compute the power spectrum of the residual series first. This is because the MLE optimization will use the power spectrum slope as the a priori value for the power-law spectral index and, in many cases, this is a reasonably good a priori value that may speed up the MLE fit.

Notes on the wavelet transform:

1. The wavelet transform is based on the inner product of the selected series and the scaled and translated complex-valued Morlet wavelet with a central frequency equal to 2π radians.

2. The user needs to set the *max/min periods*, the *period resolution*, and the *temporal resolution* (all in the same time units of the series). The max/min periods and the highest period resolution will be proposed when the user activates the wavelet option. These are based on the equivalent periodogram limits. A reasonable temporal resolution will also be initially provided, roughly corresponding to 500 epochs regularly distributed along the series length. Note that the minimum period and the temporal resolution must be set consistently in order to get a correct decomposition of the signal at high frequency.

3. The plotted gray dashed line represents the cone of influence of the transform, i.e., for long periods outside this line, edge effects may bias the transform. The solid black lines represent the 1σ, 2σ and 3σ values of magnitude (assuming they are normally distributed). The solid black asterisk represents the location of the maximum magnitude.

4. Compared to the periodogram or the KF, the maximum period estimated by the wavelet transform corresponds to half the observed series. In addition, even if results are obtained from unevenly spaced series, data gaps will still seriously affect the wavelet transform. Also, the estimated magnitude is not exactly comparable to a sinusoidal amplitude; the estimated magnitude can deviate from a sinusoidal amplitude by up to 5% (estimated empirically, further testing to be done).

5. Computing the wavelet decomposition at full resolution can be horribly time-consuming. A warning will be shown on the screen with a rough estimate of the time it may take. If the expected computation time is too long, it is recommended to start with a lower temporal and/or spectral resolution to get a generalized idea of the heat map. Then the user can iteratively improve the resolution as needed and at the same time reducing the max/min periods to focus on areas of interest in order to save time.

6. The time needed to compute the wavelet transform is commensurate with the amount of server memory used. Since the online server memory is limited and shared among simultaneous users (see details in the Known issues section), large wavelets may not finish, or even the server could kill the user's session, if there is not enough free memory. Unfortunately, this happens without any warning.

Notes on the Vondrák smoother:

1. The smoother acts as a band-pass filter and accepts two input periods: a *low-pass* and a *high-pass* cut-off. To smooth the variability between these two periods, input the cut-off period values as *low pass > high pass*. On the other hand, to smooth everything outside a selected band, swap the cut-off values and set *highpass > lowpass*.

2. If the *low-pass* or the *high-pass* period is left blank, the smoother will react as a high-pass or a low-pass filter, respectively.

3. The cutoff period of the smoother is only approximate and depends on the number of points in the series. An empirical scaling has been applied to the algorithm, but there may still be a bias between the user input cut-off period and the smoother cut-off response. Fortunately, this bias is constant for each series. Therefore, the bias, if any, can be assessed and taken into account by the user by looking at the intersection between the periodograms of the filter series and the filter residual series.

4. Since it is based on cubic splines, the quality of the Vondrák smoother may degrade very close to the data limits at the start and end of series, but also near long data gaps and discontinuities.

5. If the band to be smoothed is narrow, a significant amount of signal will remain in the series. This is due to the relatively slow "spectral response" of the smoother, which is slightly worse than a Hanning window (Sylvain Loyer, personal communication).

6. The effect of filtering the series inside/outside the specified band can be assessed by plotting the periodogram of

the filter and the filter residuals, or by using the wavelet transform.

7. If the series are fitted by an LS or KF model, the residual plot will show the differences between the smoothed series and the fitted model in yellow color. If there is no LS or KF model, the residual plot will show the smoother residuals.

Notes on the noise analysis:

1. The noise analysis is applied to the residual series from the model fit or from the Vondrák filter. In case both series exist, the priority is given to the model residuals.

2. The covariance model can be created from four stochastic processes commonly seen in GNSS position time series: white noise, flicker noise, random walk noise and power-law noise.

3. The power-law noise is a generalization of the other three noise processes and, thus, it is not possible to estimate power-law with other colored noise such as flicker or random walk, but it is compatible with white noise, though.

4. The user needs to select the components of the noise model among the four proposed and SARI will provide the more likely values of the parameters of those components (noise standard deviation and spectral index). The estimated log-likelihood can be used to assess which noise model is more likely to fit the residual series better. The power spectrum of the residual series can also be used to visually assess whether the estimated noise model is a reasonable fit of the series covariance.

5. After the MLE fit, the left panel will contain the estimated standard deviation of each noise process, given in $s\ t^{\frac{k}{4}}$ units, where s ant t are the `Series units` and the `Time units`, respectively, and k is the spectral index: *0* for white noise, *-1* for flicker noise, *-2* for random walk noise. If the `Noise uncertainty` option was selected, the estimated formal errors will also be shown. The formal errors of the noise parameters can be used to assess whether a noise component was clearly detected in the optimization process.

6. A priori values of the noise parameters are computed assuming they all contribute equally to the series variance, except for the a priori RW variance which is kept minimum. The a priori value of the spectral index for the power-law model is that of a flicker noise model, but a better a priori value can be used if the power spectrum of the residual series has been computed (see the notes above).

7. Several optimization methods have been implemented, both constrained and unconstrained. At this moment, SARI only uses the unconstrained quasi-Newton method proposed by Schnabel et al (1985), which was found to be the fastest among those tested.

8. The processing time of the *noise analysis* grows considerably for long series with high sampling and complex noise models. An estimate of the expected processing time will be shown on the screen, based on the number of points in the series and the chosen noise model. Note that the expected time is only approximate, as it depends on how well the requested noise model fits the series and on the CPU used for the computations. The expected times shown on screen are those for a typical online SARI session, where a WN+PL model fit to a fifteen-year-long daily series will take more than 10 minutes. Using SARI on a local session should produce faster noise analyses than the online sessions, where resources are shared among users, even CPU in case of many simultaneous users (see details in the Known issues section). For local sessions, compiling the source code

of the necessary R packages on a Linux machine may produce significant gains of processing time, somewhere around 10 times faster, compared to installing the package binaries on a Windows machine. In addition, for local sessions on a Windows machine, there are optional packages that, when installed, would speed up the noise analysis significantly (see details in the INSTALL instructions file). There is still room for optimization of the code in future versions, but at this moment, for online SARI sessions, it is recommended to reduce the sampling of the series to the minimum necessary. For instance, the periodogram can be used to assess the period from which the power-law noise starts dominating the variance and then adjust the sampling of the series to half that period; the white noise will be mostly averaged out. For GNSS position time series, a weekly sampling is a reasonable choice. In that case, even a WN+PL model fit (or PL only actually) to a 34-year weekly series on an online session should take less than 1 minute to finish.

9. For noise models that include more than one component, the uncertainty of the estimated parameters is obtained from a numerical approximation of the Hessian of the log-likelihood. If the series is very long, computing the Hessian numerically can take a significant amount of time, so if the uncertainties of the noise parameters are not needed, deactivating the option `Noise uncertainty` will skip this computation and speed up the noise analysis. For noise models with a single component, the Hessian of the likelihood can be obtained analytically and very quickly, so the `Noise uncertainty` option will not be available (uncertainty of the noise component computed by default).

10. The `Noise separation` option uses the Wiener filter to decompose the residual series into several noise component series following the relative variance-covariance contribution of each estimated noise component. The estimated noise series are printed to the downloaded results file.

11. The LS model fit is not updated with the estimated covariance matrix of the residuals. However, if the fitted LS model includes a linear trend, an approximation of its formal uncertainty is provided by taking into account the estimated colored noise. This is done using the general formula for the uncertainty in the rate for a power-law noise from Williams (2003). The estimated rate uncertainty from the noise analysis is then included into the linear trend uncertainty.

12. The ratio of the linear trend uncertainty (colored vs. white estimates) is also displayed on the left panel. This parameter represents a rough estimate of the amount of colored noise accounted for by the fitted noise model and can be used to compare different noise models. Note that the white noise uncertainty used in this ratio is estimated separately and does not necessarily correspond to the standard LS trend uncertainty, which includes the effect of the fitted model and especially the offsets. In other words, if the fitted LS model has many offsets, the colored noise might actually not contribute significantly to the linear trend uncertainty already provided by the LS fit.

13. If the power spectrum of the model residual or filter residual series is plotted, the power spectrum of the fitted noise model will be drawn on top of it. The crossover periods between the different noise components (if there is more than one noise component) will be shown on the left panel.

14. The estimated noise parameters will be used automatically in the `Check offsets` option to assess the statistical significance of the estimated offsets in a LS fit (see the Fit controls block).

Go to top of document

Interactive operation

The top navigation bar contains the links to this *help page* and for changing between the 3D, the different coordinate components and the residual series (unless a 1D series is used). It also contains two links at the right edge to download this help page into a `PDF` file and to `save` the results of the analysis. These two links will not be visible unless they are needed. To get back to the main program interface from this help page, click on any of the coordinate component tabs.

SARI processes the time series of each coordinate component independently. However, many options and features activated while analyzing the series in one of the tab/components will still be active when changing the tab/component (if using a 3D series) or the column number (if using a 1D series). This implies that when changing the series component, the visualization of the new series will not be available until all the selected features are computed. This allows fitting the same model for different components of the series or different series in the same input file, although it may also slow down the initial visualization of the series. It is recommended to deactivate the options that are not needed to have a faster and smoother transition between coordinate components. Also, bear in mind that a model that has been adjusted in one series may fail for another (for instance, if fitting a logarithmic/exponential decay). Since the Kalman filter is run only when clicking on the `run KF` button, the KF fit will be removed when changing the tab.

One click on any plot will provide the coordinates of the clicked point in a space right under the plots. This is useful to obtain the series values, pinpoint discontinuity epochs, or periodic lines in the spectrum. Note that in case a secondary series is plotted, the coordinates of the clicked point will correspond to the vertical axis of the primary series on the left, which very likely, will be different to the axis of the secondary series on the right, unless the `same axis` option is activated (see the [notes on the secondary series](#)).

One click and drag will select an area on any plot. Once the area is selected, its dimensions and location can be modified. One click outside the area will remove the selection. A double click inside the area of any plot will zoom in on the selection. A double click on the same plot with no selected area will zoom out (if possible) to the original view. Points inside the red selected area on the series plot (top) or the blue selected area on the model/filter residuals plot (bottom) can be excluded using the `toggle points` button. This is useful for removing outliers or for limiting the series to a period of interest (see the `truncate` option for that also). The removed points will turn into solid red on the series plot as a remainder, but they will disappear on the model/filter residual plot. If the option `permanent` is activated, the toogled points will be removed from the analysis for good (but not from the user input file). If the series has more than one dimension, the removed points for each dimension will be kept separately when changing tabs if the `all components` option is not activated.
Any point toggled twice on the series plot (top) will be restored as a valid point again, i.e., off and on, unless the the option `permanent` was used. All *toggled* points in the tab can be restored at once with the `reset toggle` button. If the `reset toggle` button is not active, it means there are no points to be restored.

When points are removed from the series after fitting a KF, the fit results will still be available for the remaining points, but these results will not represent the actual series after the points were removed. This is because the KF is updated only when the `run KF` button is clicked (see the [Fit controls](#) block). A warning asking to run the KF again will be shown on the screen.
For the same reason, when previously removed points are included back again into the series (restored) after fitting a KF, since there is no solution estimated for these points, the full KF results will be deleted and the user will have to run the KF again.

In addition to using the selected areas on the plots, large residuals can be automatically excluded using the `auto toggle` button. Large is defined by providing the absolute *residual threshold* or the absolute *normalized residual*, i.e.,

the residual divided by its own error bar if available. All the points outside the absolute thresholds will be excluded at once.

The `reset` button will erase all the plots and all the parameters from the server memory, except the pointer to the *station.info* and the *custom offset* files, which will remain ready to be used again (even if they may not appear as being uploaded, the `plot` and `list` options will still be available).

After finishing a series and before uploading a new one, it is recommended to click the `reset` button instead of refreshing the page.

Some features require intensive and time-consuming processing (*noise analysis*, *automatic offset detection*, *wavelet*). In order to save server resources and make the app always accessible to anyone, after 15 minutes without client-server interaction, the server will kill the connection, go to sleep, and may even dream of electric sheep like all androids surely do!

A warning should appear on the screen when a very long computation time is expected.

[Go to top of document]

Example of use

This is how I usually estimate the linear trend in a GNSS position time series (the steps may change depending on the series and application):

1) Upload the series file from your computer using the `browse file` button, or from a remote web server using the `Server`, `Product` and `Station` options, or via a URL query, or from a Unix-like terminal using the [sari.sh script].

2) If loading a series from your computer, set the series format and time units if they are not already set, and `Plot` the series with points (default), lines or points & lines.

3) Click on the `overview` button to open a new window with the three coordinate components and then select one of the coordinate components in the upper tabs. Move the overview window to a secondary screen and leave it open there.

4) Load and correct one of the available `plate models` (for horizontal components) and `GIA models` (for the vertical component) if necessary. The station's tectonic plate can be obtained with the `show location map` option. You can also load the parameters of your own plate and GIA models.

5) Upload a secondary series to correct the loading displacements (atmospheric, oceanic and hydrology) using the series available at the EOSTLS server.

6) If the series is very long (more than 15 years) and it has daily sampling, reduce it to weekly sampling using the `reduce sampling` option with a period of *=7/365.2425* years if the time units are years (or the equivalent 7 days or 1 week, depending on the series time unit). If the series time units are years, it will be more precise and preferable to change the time units to days or weeks, use the `reduce sampling` with a period of *7* or *1*, and then change the time units back to years if necessary.

7) Upload a previously unfinished fit or the fit from another series, if comparison is needed, with the `load SARI model` button.

8) Upload the *sitelog* and/or *station.info* and/or *soln* and/or *custom offset* files from your computer if available. The *station.info* and *custom offset* files are already loaded if you have done this and did not refresh the page.

9) Show the equipment changes by activating `plot changes` next to the input *sitelog*, *station.info*, *soln* and/or *custom* files.

10) Fit a low-pass `smoother` (a period of 0.1 years should be enough to fit seasonals) and remove the outliers from the filter residuals manually using the `toggle points` button, or automatically using a residual threshold and the `auto toggle` button. Activate the `all components` option before removing any outlier if you want to have the same points removed from the three coordinate components. If you need to keep track of the removed points, activate the `include in file` option. If any of the points is extremely far from the series and does not allow visualizing them correctly, remove them right after activating the `permanent` option.

11) Remove the low-pass smoother (optional) and fit a linear trend using weighted LS.

12) Fit position discontinuities due to known equipment changes (dates are available by activating `list` changes next to the input *sitelog*, *station.info* and/or *custom* files) or due to unknown events (dates are available under the plot by zooming in and clicking where a discontinuity is needed).

13) Alternatively, if feeling lazy today, you can also try to `search discontinuities` automatically (this is very time-consuming, so do not forget to reduce the sampling to weekly).

14) Plot the power `periodogram` of the model residuals and look for significant periodic lines.

15) Remove sinusoidal variations and check the periodogram. The exact periods can be obtained by zooming in and clicking on the periodogram. Rise the oversampling factor after zooming in if necessary, but return to 1 before zooming out. If the periodic variations are not sinusoidal, use the generic `periodic waveform` instead of the LS sinusoidal fit. If the periodic variations are sinusoidal, but their amplitude and/or phase are changing smoothly, use a Kalman filter fit with appropriate process and measurement noise variances. Temporal changes of amplitude and frequency can be assessed using the `wavelet` analysis (this is very time-consuming, so you may need to reduce the period bounds and/or reduce the temporal/frequency resolution of the wavelet).

16) Add logarithmic and/or exponential terms to the fitted model if necessary. Add also a higher-degree polynomial in the extremely unlikely case that this could be explained from a geophysical point of view.

17) Iterate the steps 11 to 15 while zooming in and out guided by the smoother, the known equipment changes, the other coordinate components shown in the overview window, an independent deformation model, or the series of a nearby station, which can be uploaded with the *secondary series* option (only if loading corrections were not already uploaded in step 4).

18) Check the Student's t-test of significance for each one of the fitted model parameters in the fit summary (bearing in mind that formal errors may be too optimistic) and remove unnecessary discontinuities from the model.

19) When the fit of the LS model is finished, run the `noise analysis` using a white + power-law noise model to get a better formal rate uncertainty (this can be very time-consuming, so do not forget to reduce the sampling to weekly if necessary).

20) Check the power `periodogram` (from step 13) to validate that the fitted noise model corresponds to the LS model residuals, or use the estimated log-likelihood to select the best noise model. The parameters of the fitted noise model will be used to estimate a more realistic formal uncertainty of the linear trend.

21) Use the estimated noise parameters to iteratively check the significance of the estimated offsets using the `check offsets` option and remove offsets from the LS model one by one if their probability is less than 95 %.

22) Plot the `histogram` if statistics of the model residuals or a stationarity assessment are needed. The standard deviation of the residuals is also indicated on the overview figure.

23) Run the `MIDAS` and/or the `minimum entropy` estimators for trend comparison.

24) Save the results of each coordinate component on your computer using the `save` icon at the top right corner. This file can be uploaded with the `load SARI model` if you need to redo the same fit later. If necessary, the periodogram data can be saved by clicking on the `get periodogram data` link below the periodogram.

25) Iterate through the different coordinate components and save each one of them. The same component can be saved multiple times, for instance if the model was improved. You can also save a full-page screenshot with all the plots and analysis results. To do that it is recommended to deactivate the `scrolling` option first.

26) When all the coordinate components are done, click on the *residuals* tab and check if anything else is missing in the fit. Check the residual series and the fitted series in the overview window side to side. If all is good, click on the `save` button to download the residual series into a file.

27) Right click on the `overview` window and save the PNG figure to be used later in a presentation or in the supplemental material of a paper.

28) Click on the `reset` button before starting a new series and enjoy!

Do not refresh the page unless you encountered a problem, in which case, I would be interested in knowing more about it.

Known issues

There is a finite number of simultaneous connections that can be run by the online server at Shinyapps. That number depends on the user load. For standard user loads (i.e., excluding noise analysis with spectral index, automatic discontinuity search, and large wavelet analysis), up to 24 parallel users can be sustained, possibly a few more than that. However, it is not very clear how many simultaneous connections the Shinyapps server allows from the same inbound IP address; that limit may be set somewhere between 10 and 15. If using SARI in a class with many people and connections start being refused, I suggest changing the outbound IP address (for instance, connecting through a smartphone hotspot) or using SARI from a local server (RStudio or Docker).

Contact the author for further information.

For the sake of simplicity of the model equation, SARI uses a common nonlinear LS function that most of the time converges in one iteration as expected. On very rare occasions, the fit does not converge, even with very simplistic models that usually include a sinusoid. A *"unable to fit the LS model. Change the model components"* message is shown on the screen. To avoid this problem, changing slightly the reference epoch of one of the model parameters should be enough to reach convergence.
Contact the author if you encounter this problem.

The Kalman filter/smoother is significantly slower than the least squares fit, at least for typical GNSS position series I have tested and especially if the model has many parameters. Between EKF and UKF, UKF is theoretically more robust against nonlinearity, though this may not represent a big difference for typical GNSS series. Therefore, UKF is the preferred Kalman flavor by default, and it is the algorithm behind the measurement noise optimization as well (see the Fit controls block).
However, I have noticed that, for some series and depending on the model being fitted, the UKF provides negative variances at some epochs for some of the estimated state parameters.
In my tests, the state itself looks good, so at this point, I am not sure what is wrong with the variances, but it may be related to rounding errors or to the unscented transformation of the selected sigma points.
The uncertainty of the estimated state parameters at the affected epochs will be set to *NA* in the downloaded file. If you encounter this problem, you can use the EKF implementation.
Also, note that EKF and UKF are not necessarily providing the same fit to the series.

At this moment, I have not found a way to select the output directory on the client's side, where the user downloads the processed series, from the online server's side to avoid the repetitive, and sometimes annoying, download prompt. This is related to server/client standard secure browsing. At least, some web browsers provide extensions to automatically download to a specific directory given the file extension or to avoid the unnecessary *(1)*, *(2)*, . . . added to the file name if downloaded several times.
If running SARI on a local server (RStudio or Docker), there is a sixth block at the bottom of the left panel, hidden for remote connections, that allows setting a directory on your machine and save the results there with just a button click (no download prompt).

Go to top of document

Acknowledgements

I'm open to any constructive comment or suggestion that could improve this software, including bug repports or missing features.
For any comment or question, please open a github issue.

If you use SARI for your research, I would really appreciate it if you could include this paper in your references.

As for me, I am thankful to these people that directly or indirectly contributed to improve this software (some of them more than once):

Valérie Ballu, Sylvain Loyer, Paul Rebischung, Pascal Gegout, Giorgi Khazaradze, Alexandre Michel, Emilie Klein, Jean-Michel Lemoine, Guy Wöppelmann, Sara Padilla, Sorin Nistor, Massyl Ouaddour, Kevin Gobron, Juan J. Portela Fernández, Marianne Métois, Andrea Walpersdorf, Germinal Gabalda, Hanane Ait-Lakbir, Florent Feriol, Médéric Gravelle, David Rodríguez Collantes, Daniel Moreira Medeiros, Elena Gimenez de Ory, Audrey Hyeans, Julie Cheynel,

Mikaël Guichard, Joëlle Nicolas, Laurent Soudarin, Guilhem Moreaux.

Go to top of document

References

Altamimi Z, Métivier L, Rebischung P, Collilieux X, Chanard K, Barnéoud J (2023) ITRF2020 plate motion model. Geophysical Research Letters, 50. doi: https://doi.org/10.1029/2023GL106373

Argus DF, Gordon RG, DeMets C (2011) Geologically current motion of 56 plates relative to the no-net-rotation reference frame. Geochemistry, Geophysics, Geosystems, 12, 11. doi: https://doi.org/10.1029/2011GC003751

Bird P (2003) An updated digital model of plate boundaries. Geochemistry, Geophysics, Geosystems, 4, 3. doi: https://doi.org/10.1029/2001GC000252

Blewitt G, Kreemer C, Hammond WC, Gazeaux J (2016) MIDAS robust trend estimator for accurate GPS station velocities without step detection. J Geophys Res Solid Earth 121(3):2054-2068. doi: https://doi.org/10.1002/2015JB012552

Brent RP (1971) An algorithm with guaranteed convergence for finding a zero of a function. Comput J 14(4):422-425. doi: https://doi.org/10.1093/comjnl/14.4.422

Caron L, Ivins ER (2019) A baseline Antarctic GIA correction for space gravimetry. Earth and Planetary Science Letters, 115957. doi: https://doi.org/10.1016/j.epsl.2019.115957

Eyer L, Bartholdi P (1999) Variable stars: Which Nyquist frequency? Astron Astrophys Suppl Ser 135, 1–3. doi: https://doi.org/10.1051/aas:1999102

Julier SJ, Uhlmann JK (2004) Unscented filtering and nonlinear estimation. Proc IEEE 92(3):401-422. doi: https://doi.org/10.1109/JPROC.2003.823141

Keitt TH (2008) Coherent ecological dynamics induced by large-scale disturbance. Nature 454:331-334. doi: https://doi.org/10.1038/nature06935

Kreemer C, Blewitt G, Klein EC (2014) A Geodetic Plate Motion and Global Strain Rate Model. Geochemistry, Geophysics, Geosystems, 15, 10. doi: https://doi.org/10.1002/2014GC005407

RESIF (2017) RESIF-RENAG French National Geodetic Network. RESIF - Réseau Sismologique et géodésique Français. doi: https://doi.org//10.15778/resif.rg

Peltier WR, Argus DF, Drummond R (2018) Comment on "An assessment of the ICE-6G_C (VM5a) glacial isostatic adjustment model" by Purcell et al. J Geophys Res Solid Earth, 123. doi:https://doi.org/10.1002/2016JB013844

Purcell A, Tregoning P, Dehecq A (2016) An assessment of the ICE6G_C(VM5a) glacial isostatic adjustment model. J. Geophys. Res. Solid Earth, 121. doi:https://doi.org/10.1002/2015JB012742

Saleh J, Bennett RA, Williams SDP (2024) Minimum-entropy velocity estimation from GPS position time series. J Geod 98:11. doi: https://doi.org/10.1007/s00190-023-01820-3

Särkkä S (2008) Unscented Rauch–Tung–Striebel Smoother. IEEE Transactions on Automatic Control, 53(3):845-849. doi: https://doi.org/10.1109/TAC.2008.919531

Schnabel RB, Koontz JE, Weiss BE (1985) A modular system of algorithms for unconstrained minimization. ACM Transactions on Mathematical Software, 11. https://doi.org/10.1145/6187.6192

Vondrák J (1977) Problem of Smoothing Observational Data II. Bull Astron Inst Czechoslov 28:84

Williams SDP (2003) The effect of coloured noise on the uncertainties of rates estimated from geodetic time series. J Geod 76(9-10):483-494. doi: https://doi.org/10.1007/s00190-002-0283-4

Zeileis A, Kleiber C, Krämer W, Hornik K (2003) Testing and Dating of Structural Changes in Practice, Computational Statistics and Data Analysis, 44, 109-123. doi: https://doi.org/10.1016/S0167-9473(03)00030-6

Go to top of document

R packages

Attali D (2021) shinyjs: Easily Improve the User Experience of Your Shiny Apps in Seconds. R package version 2.1.0. https://CRAN.R-project.org/package=shinyjs

Azzalini A, Genz A (2022) The R package 'mnormt': The multivariate normal and 't' distributions (version 2.1.1). http://azzalini.stat.unipd.it/SW/Pkg-mnormt/

Bailey E (2022) shinyBS: Twitter Bootstrap Components for Shiny. R package version 0.61.1. https://CRAN.R-project.org/package=shinyBS

Barrett T, Dowle M, Srinivasan A, Gorecki J, Chirico M, Hocking T (2024) data.table: Extension of data.frame. R package version 1.15.4. https://CRAN.R-project.org/package=data.table

Bates D, Eddelbuettel D (2013). Fast and Elegant Numerical Linear Algebra Using the RcppEigen Package. Journal of Statistical Software, 52(5), 1-24. doi: 10.18637/jss.v052.i05 (URL: https://doi.org/10.18637/jss.v052.i05).

Bengtsson H (2024) matrixStats: Functions that Apply to Rows and Columns of Matrices (and to Vectors). R package version 1.3.0. https://CRAN.R-project.org/package=matrixStats

Borchers HW (2023) pracma: Practical Numerical Math Functions. R package version 2.4.4. https://CRAN.R-project.org/package=pracma

Chamberlain S, Teucher A, Mahoney M (2023) geojsonio: Convert Data from and to 'GeoJSON' or 'TopoJSON'. R package version 0.11.3. https://CRAN.R-project.org/package=geojsonio

Chang W (2021) shinythemes: Themes for Shiny. R package version 1.2.0. https://CRAN.R-project.org/package=shinythemes

Chang W, Cheng J, Allaire JJ, Sievert C, Schloerke B, Xie Y, Allen J, McPherson J, Dipert A, Borges B (2024) shiny: Web Application Framework for R. R package version 1.8.1.1. https://CRAN.R-project.org/package=shiny

Cheng J, Schloerke B, Karambelkar B, Xie Y (2024) leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library. R package version 2.2.2. https://CRAN.R-project.org/package=leaflet

Eddelbuettel D, Francois R, Allaire J, Ushey K, Kou Q, Russell N, Ucar I, Bates D, Chambers J (2023). Rcpp: Seamless R and C++ Integration. R package version 1.0.11, <URL: https://CRAN.R-project.org/package=Rcpp>.

Eddelbuettel D, Francois R, Bates D, Ni B, Sanderson C (2024). RcppArmadillo: 'Rcpp' Integration for the 'Armadillo' Templated Linear Algebra Library. R package version 14.0.0-1, <URL: https://CRAN.R-project.org/package=RcppArmadillo>.

Gilbert P, Varadhan R (2019) numDeriv: Accurate Numerical Derivatives. R package version 2016.8-1.1. https://CRAN.R-project.org/package=numDeriv

Grolemund G, Wickham H (2011) Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. https://www.jstatsoft.org/v40/i03/

Keitt TH (2019) mvcwt: Wavelet Analysis of Multiple Time Series. R package version 1.3.1. https://CRAN.R-project.org/package=mvcwt

Milton Bache S, Wickham H (2022) magrittr: A Forward-Pipe Operator for R. R package version 2.0.3. https://CRAN.R-project.org/package=magrittr

Neuwirth E (2022) RColorBrewer: ColorBrewer Palettes. R package version 1.1-3. https://CRAN.R-project.org/package=RColorBrewer

Nychka D, Furrer R, Paige J, Sain S (2021) fields: Tools for spatial data. R package version 15.2. https://github.com/dnychka/fieldsRPackage

Ooms J (2014) The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects. arXiv:1403.2805. https://arxiv.org/abs/1403.2805.

Petris G (2010) An R Package for Dynamic Linear Models. Journal of Statistical Software, 36(12), 1-16. https://www.jstatsoft.org/v36/i12/.

Petris, Petrone, and Campagnoli (2009) Dynamic Linear Models with R. Springer.

Revelle W (2024) psych: Procedures for Psychological, Psychometric, and Personality Research. Northwestern University, Evanston, Illinois. R package version 2.4.3. https://CRAN.R-project.org/package=psych.

Sali A, Attali D (2020) shinycssloaders: Add Loading Animations to a 'shiny' Output While It's Recalculating. R package version 1.0.0. https://CRAN.R-project.org/package=shinycssloaders

Seilmayer M (2021) spectral: Common Methods of Spectral Data Analysis. R package version 2.0. https://CRAN.R-project.org/package=spectral

Temple Lang D (2023) RCurl: General Network (HTTP/FTP/. . .) Client Interface for R. R package version 1.98-1.12. https://CRAN.R-project.org/package=RCurl

Temple Lang D (2023) XML: Tools for Parsing and Generating XML Within R and S-Plus. R package version 3.99-0.14. https://CRAN.R-project.org/package=XML

Trapletti A, Hornik K (2024) tseries: Time Series Analysis and Computational Finance. R package version 0.10-56. https://CRAN.R-project.org/package=tseries

Xie Y, Allaire JJ, Horner J (2024) markdown: Render Markdown with 'commonmark'. R package version 1.13. https://CRAN.R-project.org/package=markdown

Zeileis A, Leisch F, Hornik K, Kleiber C (2002) strucchange: An R Package for Testing for Structural Change in Linear Regression Models. Journal of Statistical Software, 7(2), 1-38. doi: 10.18637/jss.v007.i02. https://doi.org/10.18637/jss.v007.i02.

Author

This software is developed and is maintained by

Alvaro Santamaría

Centre National d'Etudes Spatiales

Geosciences Environnement Toulouse

Université de Toulouse, CNES, CNRS, IRD, UPS

Observatoire Midi-Pyrénées

Toulouse

For any comments, suggestions, questions, bugs, unexpected crashes or missing features, please open a github issue here.

Go to top of document

Privacy policy

- When using SARI via the online server at Shinyapps.io:
  The app uses essential first-party cookies to collect aggregated data on the amount of server resources (memory and duration) being used. These data are necessary for billing purposes of the online hosting service.
  The online server logs contain technical information on the users' workflow, like the activation of some of the options and functions, which allows developping the app operation, for instance, in case of error. This information is not shared with third-parties and fully respects the anonymity of users and their own data.

  SARI also uses the open-source Matomo web analytics software to collect information about the approximate geographic location (country and city), device type, operating system, browser, language and screen size. The IP, user ID and device ID are anonymized, which means that no personal information is collected from users. Both the Matomo software and its database are under full control on EU-based private servers and follow the guidelines set by the National Commission on Informatics and Liberty to comply with the current European General Data Protection Regulation.
  Users can opt-out of the Matomo tracking by setting the "Do Not Track" (or Global Privacy Control) option in their web browser.

  No third-party/non-essential cookies or cross-domain tracking are used by SARI. However, the app contains links to third-party websites. In order to access third-party content from SARI, the user may be asked to accept the terms and conditions, including cookie policies, from the external sites.

- When using SARI from a local session (RStudio or Docker):
No personal information is collected or tracked whatsoever. Only aggregated information on the number of GitHub/Docker downloads is obtained from these external sites.