santaR Graphical User Interface

Arnaud Wolfer

The `santaR` package is designed for the detection of significantly altered time trajectories between study groups, in short time-series. The graphical user interface implements all of `santaR`'s functions.

The GUI is to be preferred to understand the methodology, select the best parameters on a subset of the data before running the command line, or to visually explore results.

This vignette will:

- Detail the step-by-step use of the graphical user interface using an example dataset.

Example Data

This vignette employ the `.csv` and `.RData` files generated from `acuteInflammation` in the vignette how to prepare input data for `santaR`.

Getting Started

The graphical user interface is started as follow:

```r
library(santaR)
santaR_start_GUI(browser = TRUE)
# To exit press ESC in the command line
```

The graphical interface is divided in 4 main tabs, Import, DF search, Analysis and Export.
Import

The first input format is a .csv file containing as **rows** the observations (samples) and as **columns** the variables as well as metadata.
Columns corresponding to metadata are selected: the metadata describe the individual ID and collection time corresponding to each observation, with optionally class information for identification of inter-class differential trajectories.
Additionally, data previously imported as well as fitting results (in `.RData` format) can be loaded for further analysis or plotting (see the Export section for more details).
DF Search (optional)

Note:
The single parameter to be set by the user is the number of degrees of freedom (df) to fit the spline model. The df controls how closely the curve models the input data-points. Once the df is chosen for a dataset (a given number of time-points and missing values), it can be kept constant whichever the question to investigate (the metadata and group comparison). Some indications based on simulated data and diverse datasets can guide the selection of df:

- **df** controls the “complexity” of the model employed. A substantial difference can be found when going from 2 to 10, but very little change will take place when going from 10 to 50 (the model only gets more complex, but the general shape won’t change).

- More time points do not automatically require a higher **df**. More inflexions (more complex shape) could require a higher **df** if the number of points is sufficient (and the sampling frequency high).

- A lower **df** value is often more suited and generalisable (less over-fitted).

- If the **df** is for example 10, all individuals trajectories with less of 10 time-points cannot be fitted and will be rejected.

- On simulated data, the results (p-values) are resilient to most values of **df**, however the plots can look dramatically different.

- Trying multiple values of **df** on a subset of variables (using the GUI) and then selecting the fit that approximate the time evolution the best without over-fitting:
  - df = 5 is a good starting point in most cases (even more so if there is less than 10 time-points)
  - If the number of time-points is large and the curves seem very under-fitted, **df** can be increased to 6, 7 or more. Values higher than 10 should rarely be required and will provide with a diminishing return. df=number of time-points will result in a curve passing through all points (over-fitted).
  - If the number of points is lower or the trajectories seem over-fitted, **df** can be decreased to 4 or 3. (3 will be similar to a second degree polynomial, while 2 will be a linear model)
  - If the plots “look right” and don’t seem to “invent” information between measured data-points, the **df** is close to optimal.

It does not seem to be possible to automatically select the degree of freedom. A choice based on visualisation of the splines while being careful with over-fitting, keeping in mind the “expected” evolution of the underlying process seems the most reasonable approach.

Even if automated approaches cannot reliably select a number of degree of freedom to em-
ploy, DF search implements some of these approaches and multiple tools to help guide optimal $df$ selection.
Auto-Fit uses principal component analysis (PCA) to extract latent trajectories and generate eigen-trajectories that are subsequently assessed for optimal $df$ using various goodness-of-fit metrics.

About DF search

The single parameter to be set by the user is the number of degrees of freedom ($df$) to employ. The $df$ parameter controls how closely the curve (spline) fits the input data. It is necessary to ensure that the curve is not over-fitting or under-fitting the data. This parameter is dependent on the study design (number of time-points, sampling rate, time-scale of the function under study) and therefore only needs to be selected once per dataset.

Some indications based on simulated data and diverse datasets can guide the selection of $df$:

- $df$ controls the ‘complexity’ of the model employed. A substantial difference can be found when going from 2 to 10, but very little change will take place when going from 10 to 50 (the model only gets more complex, but the general shape won’t change).
- More time points do not automatically require a higher $df$. More inflexions (more complex shape) could require a higher $df$ if the number of points is sufficient (and the sampling frequency high).
- A lower $df$ value is often more suited and generalisable (less over-fitted).
- If the $df$ is for example 10, all individuals trajectories with less of 10 time-points cannot be fitted and will be rejected.
- On simulated data, the results ($p$-values) are resilient to most values of $df$, however the plots can look dramatically different.
- Try multiple values of $df$ on a subset of variables (using the GUI) and then select the fit that approximate the time evolution the best without over-fitting:
  - $df$ is a reasonable starting point in most cases (even more so if there less than 10 time-points are available).
  - If the number of time-points is large and the curves seem very under-fitted, $df$ can be increased to 6, 7 or more. Values higher than 10 should rarely be required and will provide with a diminishing return. $df$ number of time-points will result in a curve passing through all points (over-fitted).
  - If the number of time points is lower or the trajectories seem over-fitted, $df$ can be decreased to 4 or 3. (3 will be similar to a second degree polynomial, while 2 will be a linear model).
  - If the plots “looks right” and don’t seem to “invent” information between measured data-points, the $df$ is close to optimal.

While it does not seem to fully automate the selection of the number of degrees of freedom, DF Search implements visualisation approaches to assist in the selection of an adequate $df$ to apply across all variables for a given dataset:

Eigen-trajectories estimation

- A PCA extracts the eigen-trajectories across all variables. The $df$ that will best fit that subset of eigen-trajectories is expected to be satisfactory for all trajectories in the dataset.
- Set parameters on the left and press the run button to generate the eigen-trajectories.

Auto-Fit

- Auto-Fit returns the optimal $df$ based on different goodness of fit metrics.

Parameter Evolution

- Plot the evolution of different goodness of fit metrics for all possible $df$.
- Set parameters and press the update step button to calculate the metrics.

Plot Fit

- Plot eigen-trajectories fitted with a selected $df$ (left panel).
- Set parameters (automatic fitted spline in red) and press the update button to generate the plot.
Parameter evolution plots the evolution of these metrics across the range of possible $df$ for each latent trajectory.
To select the most suitable $df$ parameter, Plot fit generates a visualisation of the fit on each latent projection at automatically and manually selected $df$ values.

Finally **Missing value** highlight the number of trajectories that would have to be excluded as they contain less time-points than the $df$ selected.
Analysis

With the data imported and a pertinent $df$ value selected, Analysis regroups the fitting, visualisation and identification of variables significantly altered between groups.

Fit handles parameter selection as well as downstream computation. Calculation of inter-group differential evolutions can be performed with either initial class information or an advance option generated new grouping (e.g., including / combining / excluding input groups). The user can control the number of permutations and bootstrap rounds for significance and group mean curve confidence band calculation. The sub-sampling or the area between group mean curves can be altered to favour calculation speed at the expense of numerical precision. Parallelisation enables the selection of the number of CPU cores to employ for computation. View Input presents the dataset as fitted.
Plot enables the interactive visualisation of the raw data points, individual trajectories, group mean curves and confidence bands for all variables, which subsequently can be saved as an image figure to disk.
If inter-group differential evolution has been characterised, **P-value** summarise in tables all significance testing - providing multiple options for false discovery correction (e.g., Benjamini-Hochberg, Benjamini-Yekutieli and Bonferroni) as well as confidence intervals on the $p$-values.
Export

The Export tab manages the saving of results and automated reporting. Fitted data is saved as a spline object, which contains all inputs and outputs, and subsequently downloaded as .RData file for future analysis, reproduction, or analysis of results.

.csv files containing significance testing results can also be generated and summary plot for each significantly altered variable automatically saved to disk for rapid evaluation.
Final Note

If a very high number of variables is to be processed, *santaR*'s command line functions are more efficient, as they can be integrated in scripts and the reporting automated.

See Also

- Getting Started with santaR
- How to prepare input data for santaR
- santaR theoretical background
- Automated command line analysis
- Plotting options
- Selecting an optimal number of degrees of freedom
- Advanced command line options