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 prefered 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:

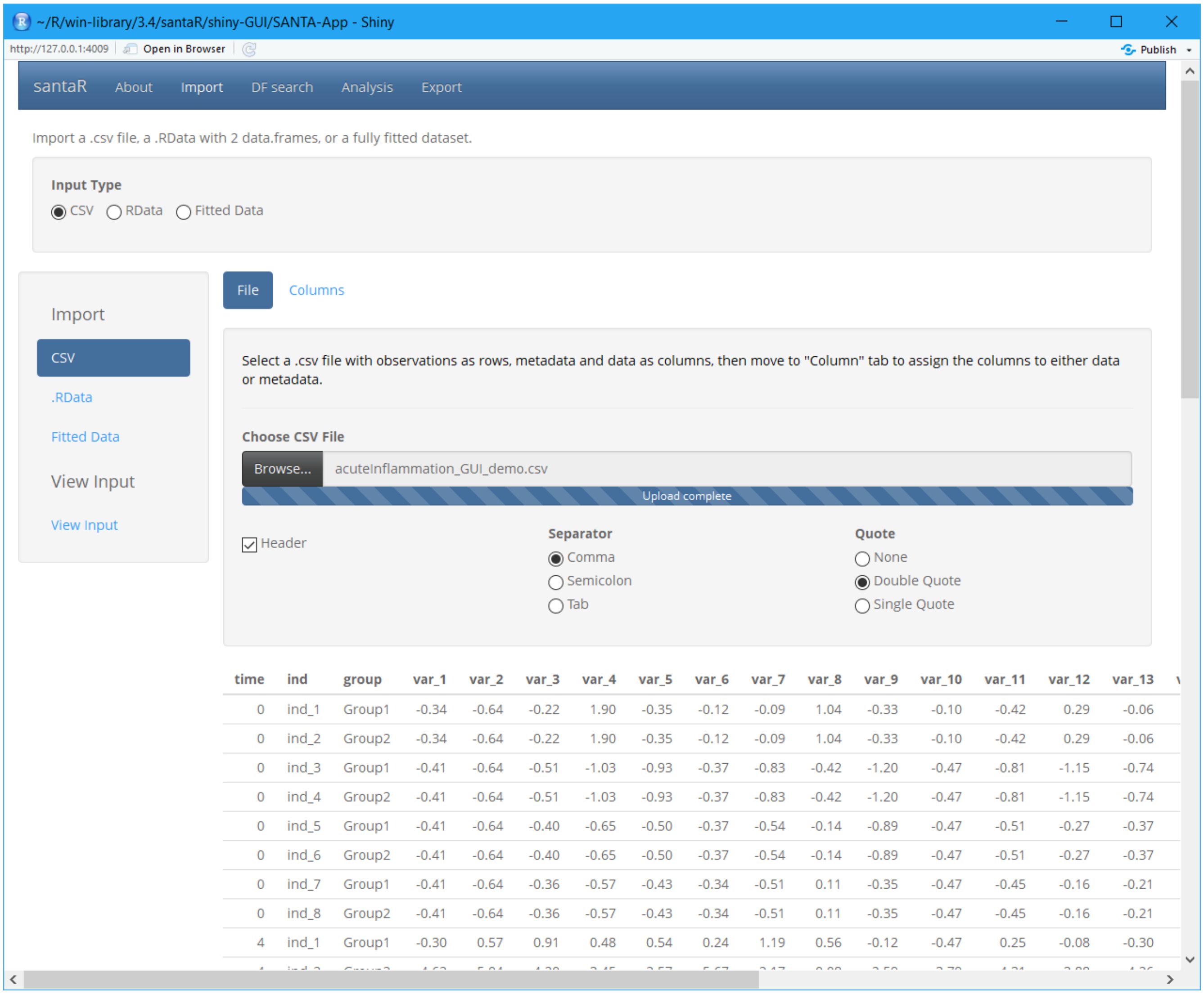
```r
library(santaR)

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.



Import a .csv file, a .RData with 2 data.frames, or a fully fitted dataset.

Input Type
● CSV ○ RData ○ Fitted Data

Select a .csv file with observations as rows, metadata and data as columns, then move to "Column" tab to assign the columns to either data or metadata.

Choose CSV File
acuteInflammation_GUI_demo.csv
Upload complete

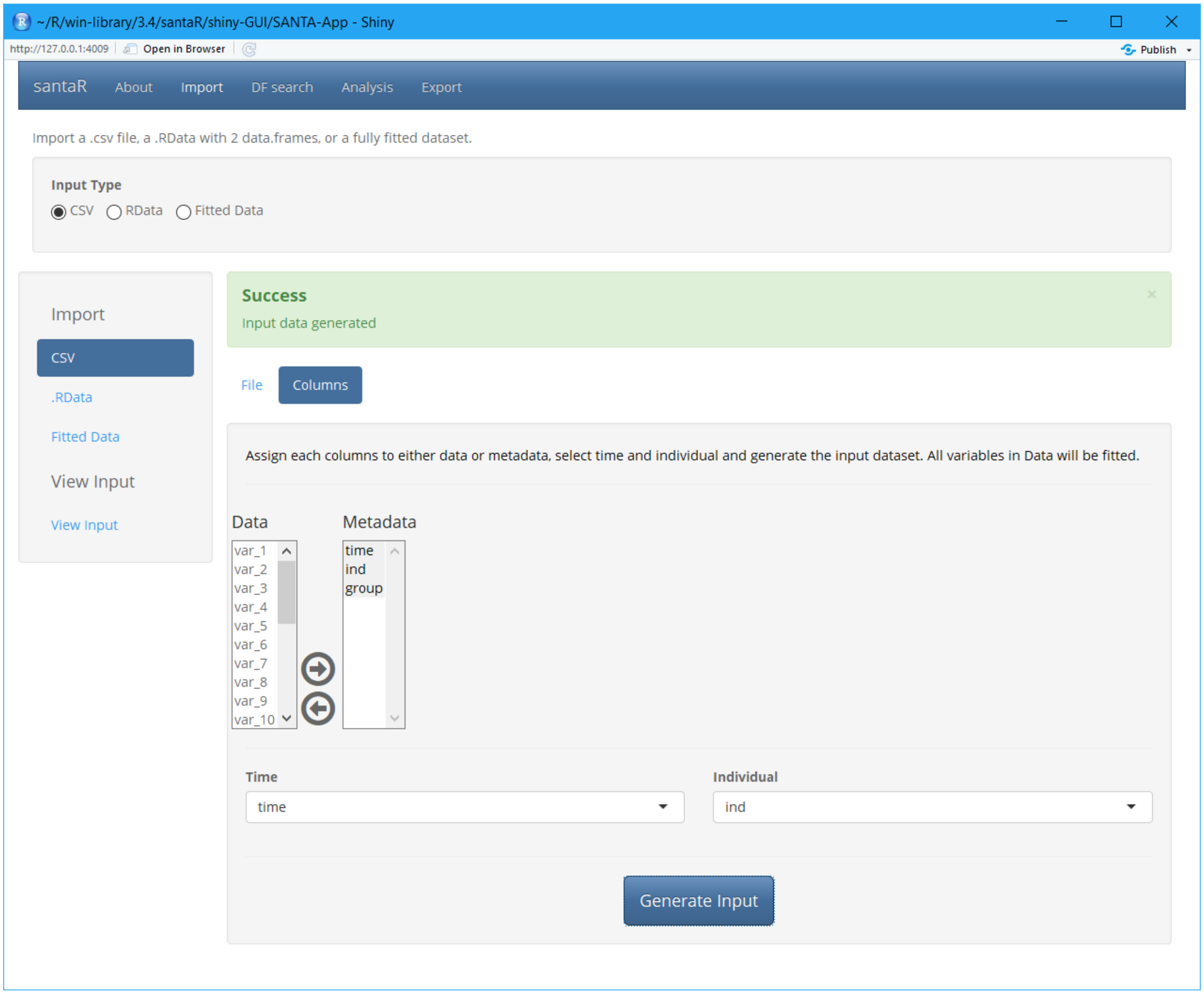
☑ Header

Separator
● Comma
○ Semicolon
○ Tab

Quote
○ None
● Double Quote
○ Single Quote

time	ind	group	var_1	var_2	var_3	var_4	var_5	var_6	var_7	var_8	var_9	var_10	var_11	var_12	var_13
0	ind_1	Group1	-0.34	-0.64	-0.22	1.90	-0.35	-0.12	-0.09	1.04	-0.33	-0.10	-0.42	0.29	-0.06
0	ind_2	Group2	-0.34	-0.64	-0.22	1.90	-0.35	-0.12	-0.09	1.04	-0.33	-0.10	-0.42	0.29	-0.06
0	ind_3	Group1	-0.41	-0.64	-0.51	-1.03	-0.93	-0.37	-0.83	-0.42	-1.20	-0.47	-0.81	-1.15	-0.74
0	ind_4	Group2	-0.41	-0.64	-0.51	-1.03	-0.93	-0.37	-0.83	-0.42	-1.20	-0.47	-0.81	-1.15	-0.74
0	ind_5	Group1	-0.41	-0.64	-0.40	-0.65	-0.50	-0.37	-0.54	-0.14	-0.89	-0.47	-0.51	-0.27	-0.37
0	ind_6	Group2	-0.41	-0.64	-0.40	-0.65	-0.50	-0.37	-0.54	-0.14	-0.89	-0.47	-0.51	-0.27	-0.37
0	ind_7	Group1	-0.41	-0.64	-0.36	-0.57	-0.43	-0.34	-0.51	0.11	-0.35	-0.47	-0.45	-0.16	-0.21
0	ind_8	Group2	-0.41	-0.64	-0.36	-0.57	-0.43	-0.34	-0.51	0.11	-0.35	-0.47	-0.45	-0.16	-0.21
4	ind_1	Group1	-0.30	0.57	0.91	0.48	0.54	0.24	1.19	0.56	-0.12	-0.47	0.25	-0.08	-0.30

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.



Import a .csv file, a .RData with 2 data.frames, or a fully fitted dataset.

Input Type

● CSV ○ RData ○ Fitted Data

Success

Input data generated

Assign each columns to either data or metadata, select time and individual and generate the input dataset. All variables in Data will be fitted.

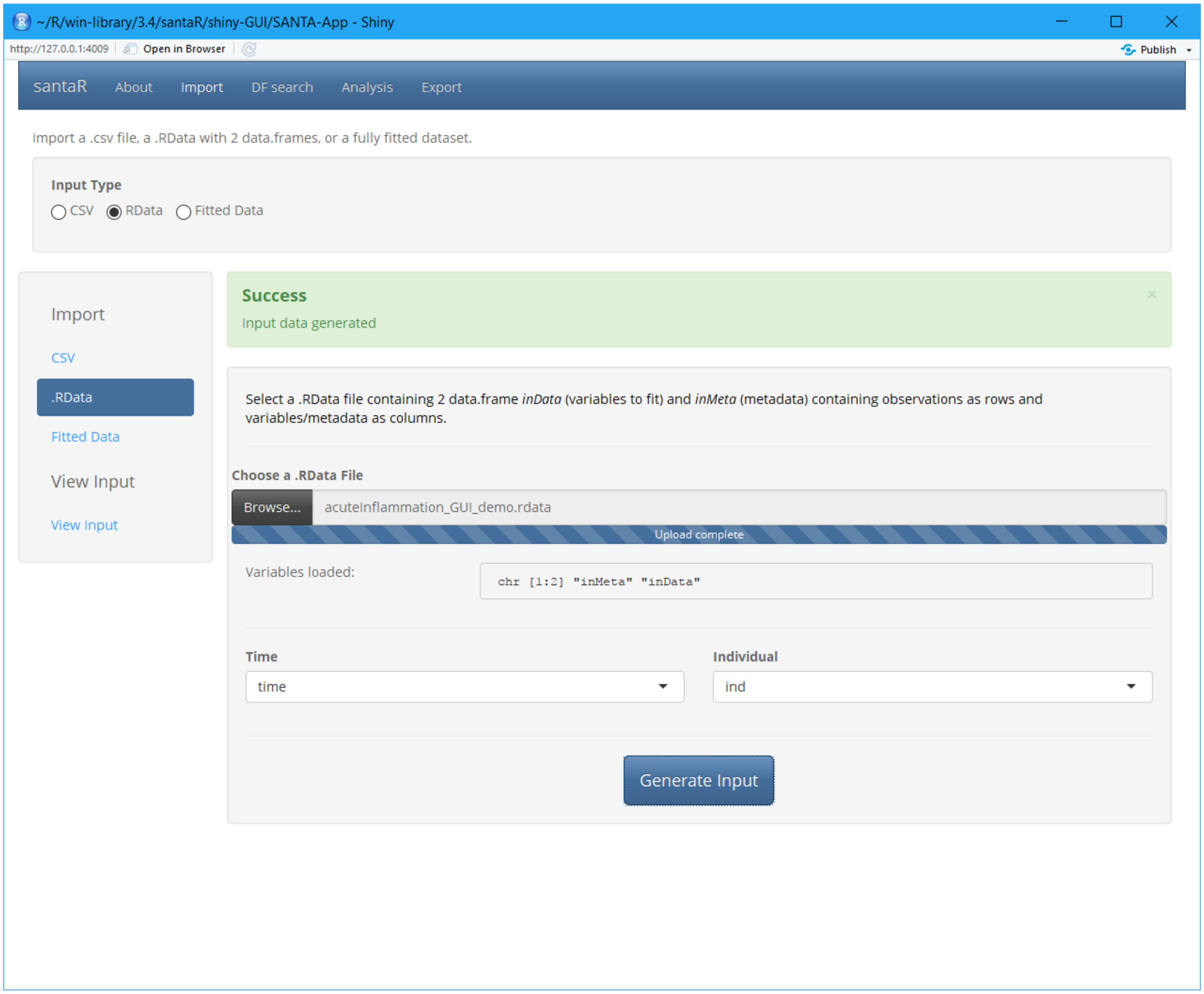
Data: var_1, var_2, var_3, var_4, var_5, var_6, var_7, var_8, var_9, var_10

Metadata: time, ind, group

Time: time

Individual: ind

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).



Import a .csv file, a .RData with 2 data.frames, or a fully fitted dataset.

Input Type

○ CSV ● RData ○ Fitted Data

Success

Input data generated

Select a .RData file containing 2 data.frame *inData* (variables to fit) and *inMeta* (metadata) containing observations as rows and variables/metadata as columns.

Choose a .RData File

Browse... acuteInflammation_GUI_demo.rdata

Upload complete

Variables loaded:

```
chr [1:2] "inMeta" "inData"
```

Time: time

Individual: ind

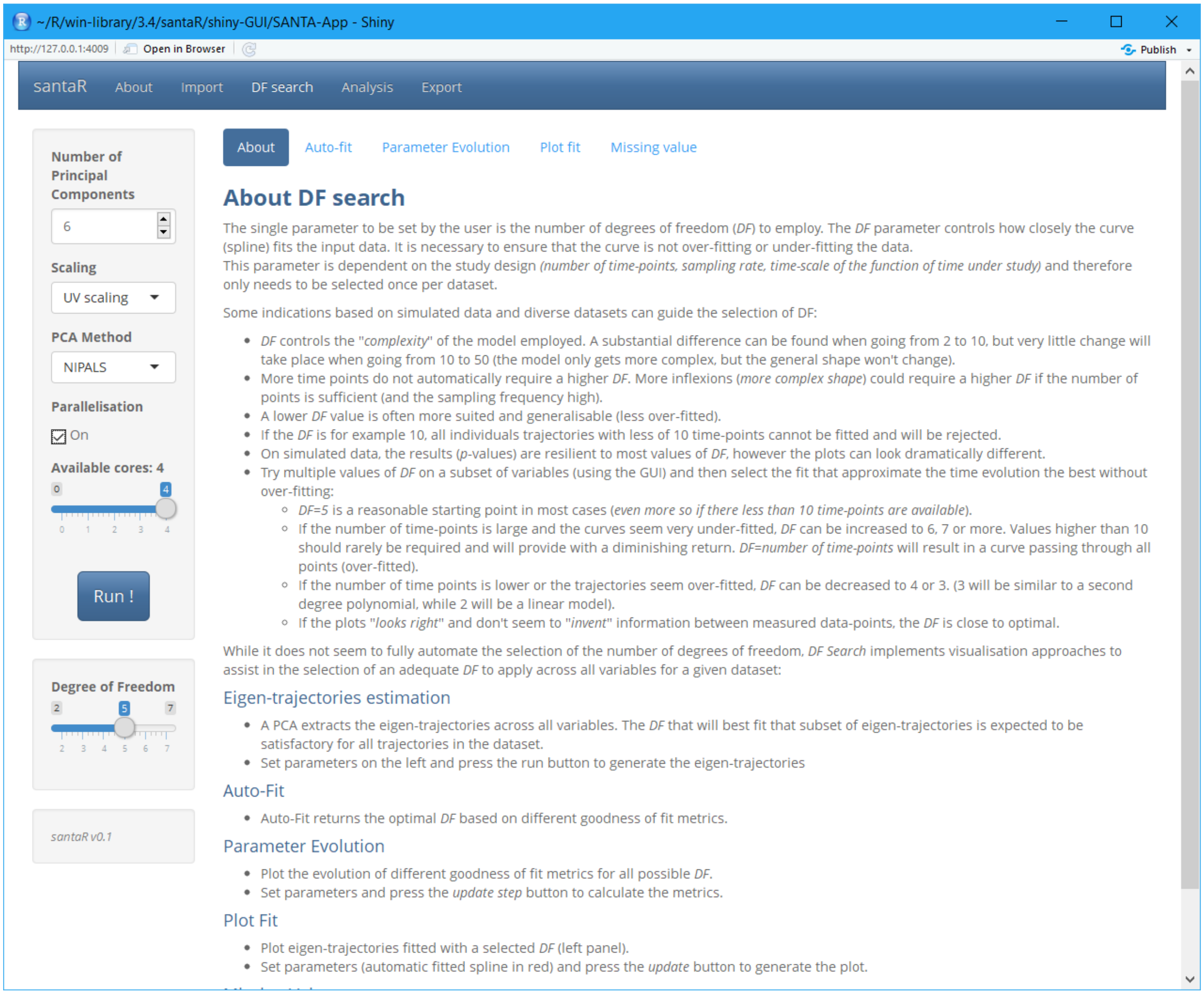
DF Search (optional)

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 employ, **DF search** implements some of these approaches and multiple tools to help guide optimal df selection.



Number of Principal Components: 6

Scaling: UV scaling

PCA Method: NIPALS

Parallelisation: ☑ On

Available cores: 4

Degree of Freedom: 5

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 of time 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=5* 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.

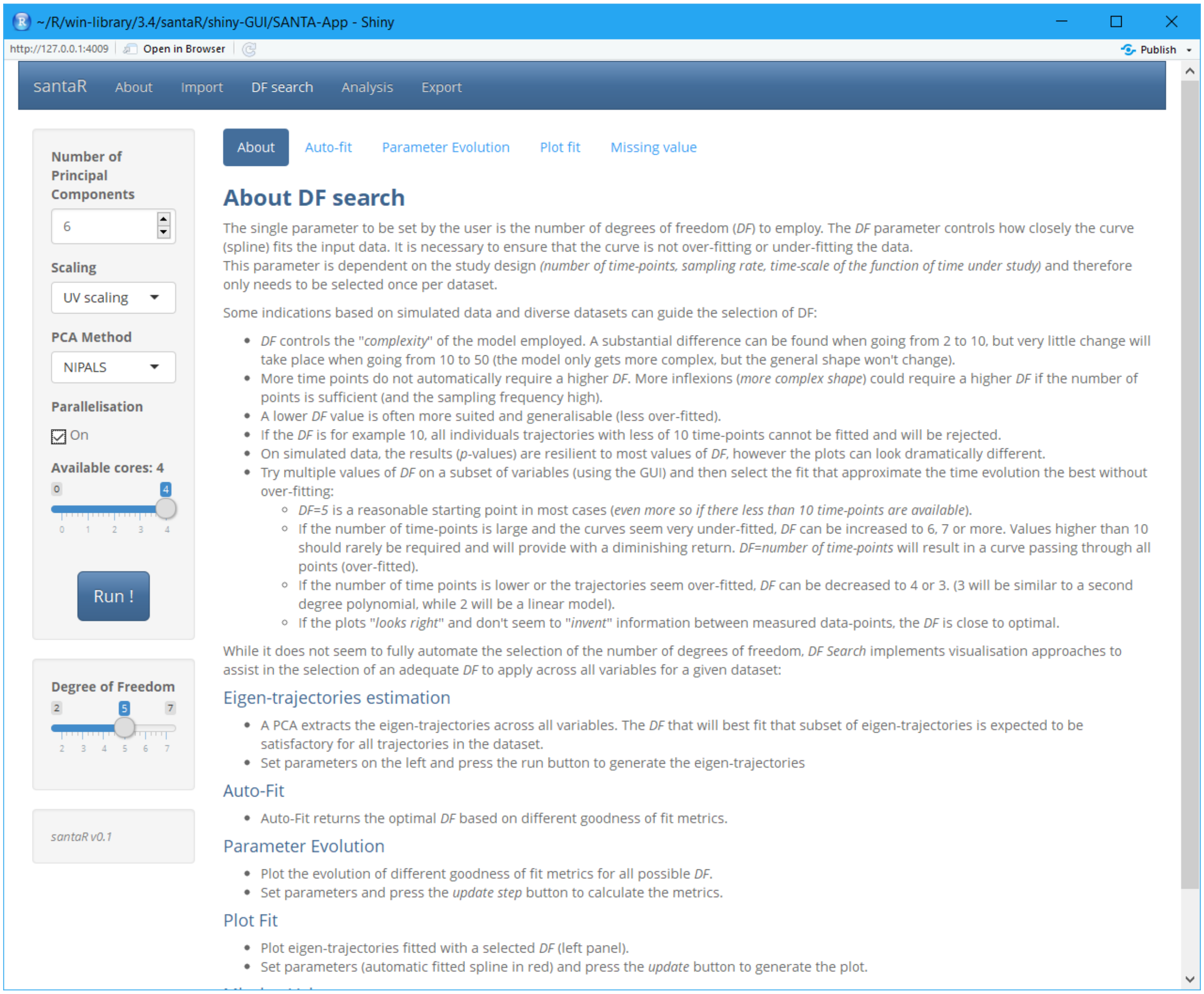
Parameter Evolution

- 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.

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.



Number of Principal Components: 6

Scaling: UV scaling

PCA Method: NIPALS

Parallelisation: ☑ On

Available cores: 4

Degree of Freedom: 5

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 of time 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=5* 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.

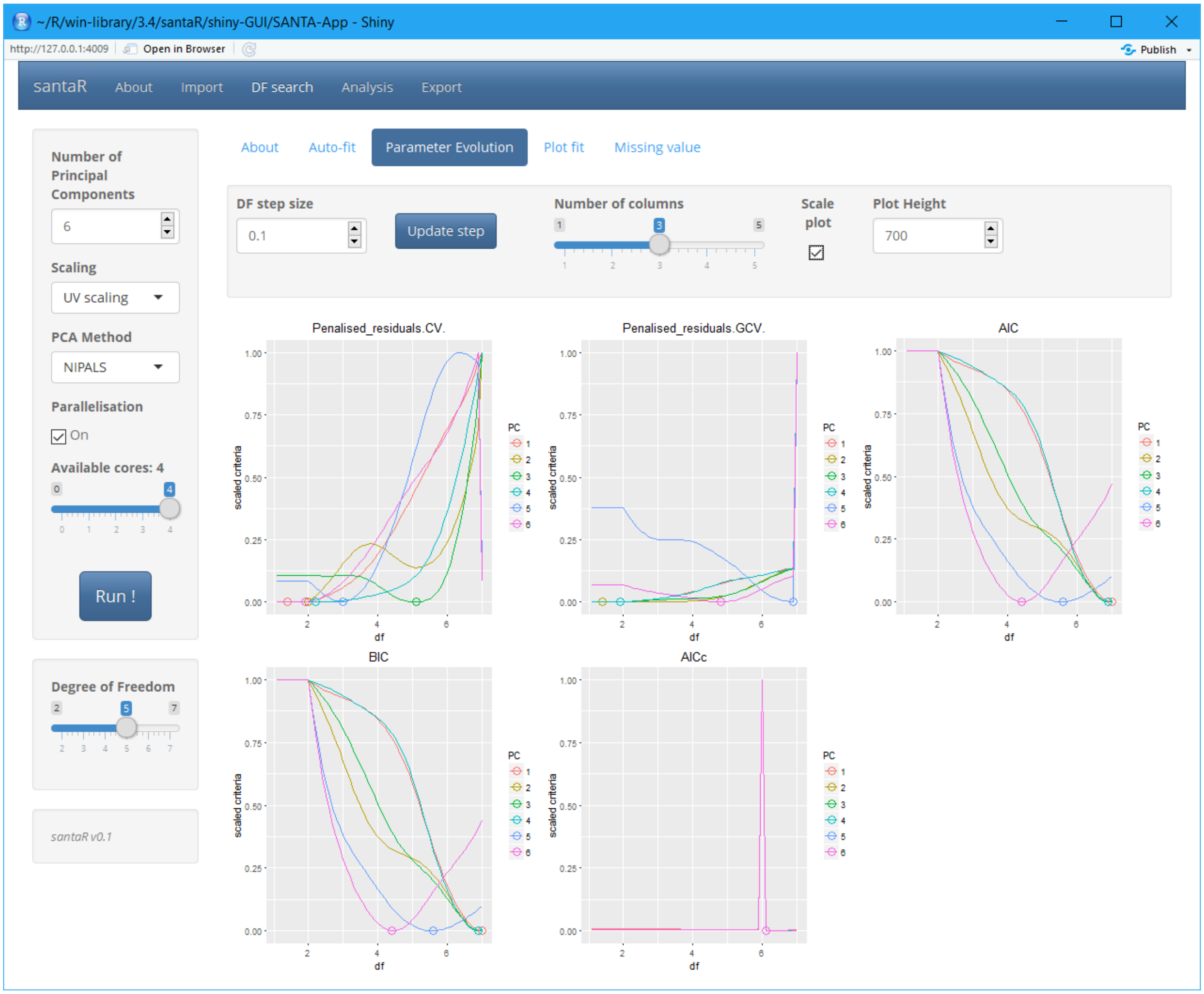
Parameter Evolution

- 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.



~/R/win-library/3.4/santaR/shiny-GUI/SANTA-App - Shiny

http://127.0.0.1:4009

Number of Principal Components: 6

Scaling: UV scaling

PCA Method: NIPALS

Parallelisation: ☑ On

Available cores: 4

Degree of Freedom: 5

DF step size: 0.1

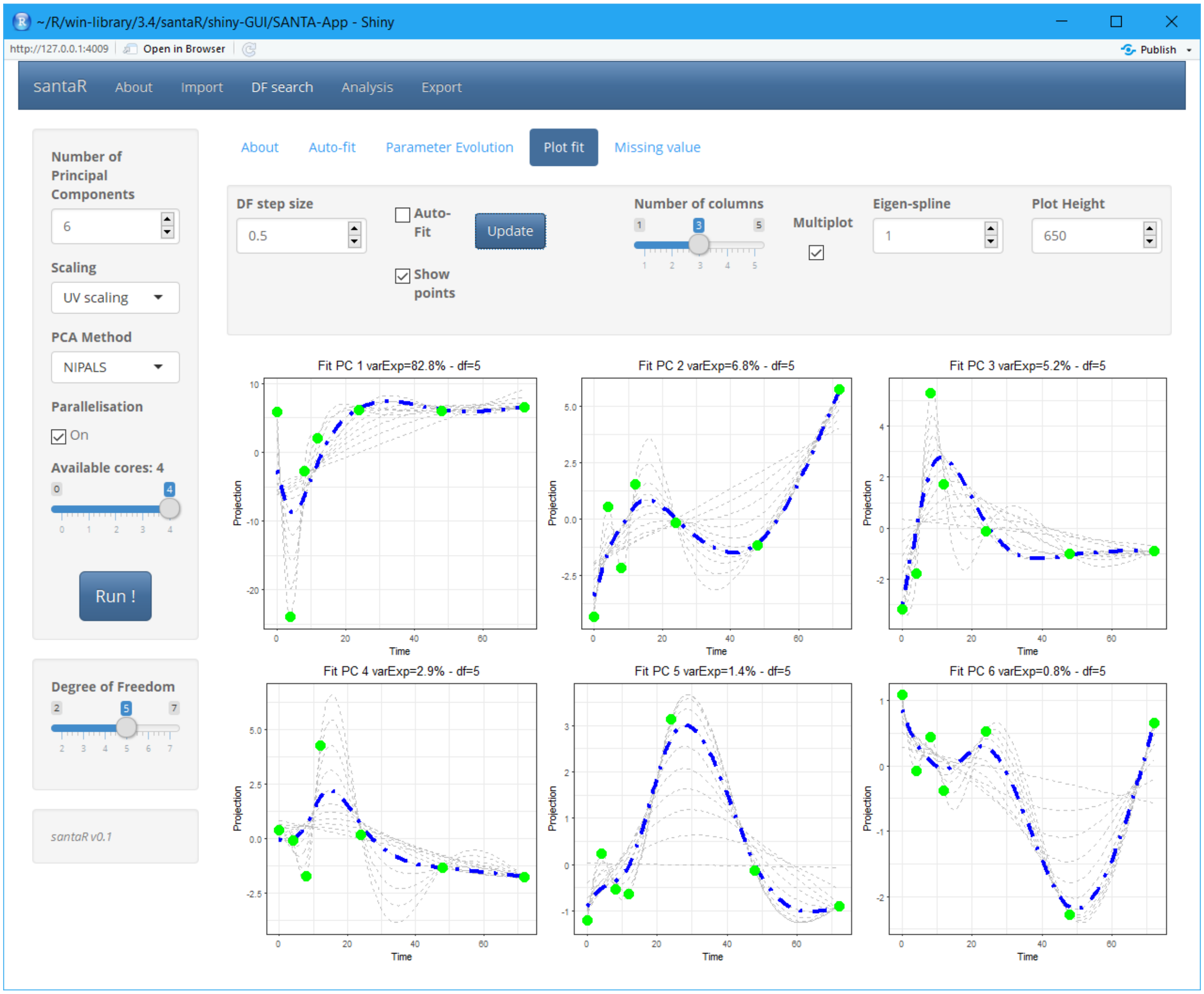
Number of columns: 3

Scale plot: ☑

Plot Height: 700

santaR v0.1

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.


~/R/win-library/3.4/santaR/shiny-GUI/SANTA-App - Shiny

http://127.0.0.1:4009

Number of Principal Components: 6

Scaling: UV scaling

PCA Method: NIPALS

Parallelisation: ☑ On

Available cores: 4

Degree of Freedom: 5

DF step size: 0.5

□ Auto-Fit

☑ Show points

Number of columns: 3

Multiplot: ☑

Eigen-spline: 1

Plot Height: 650

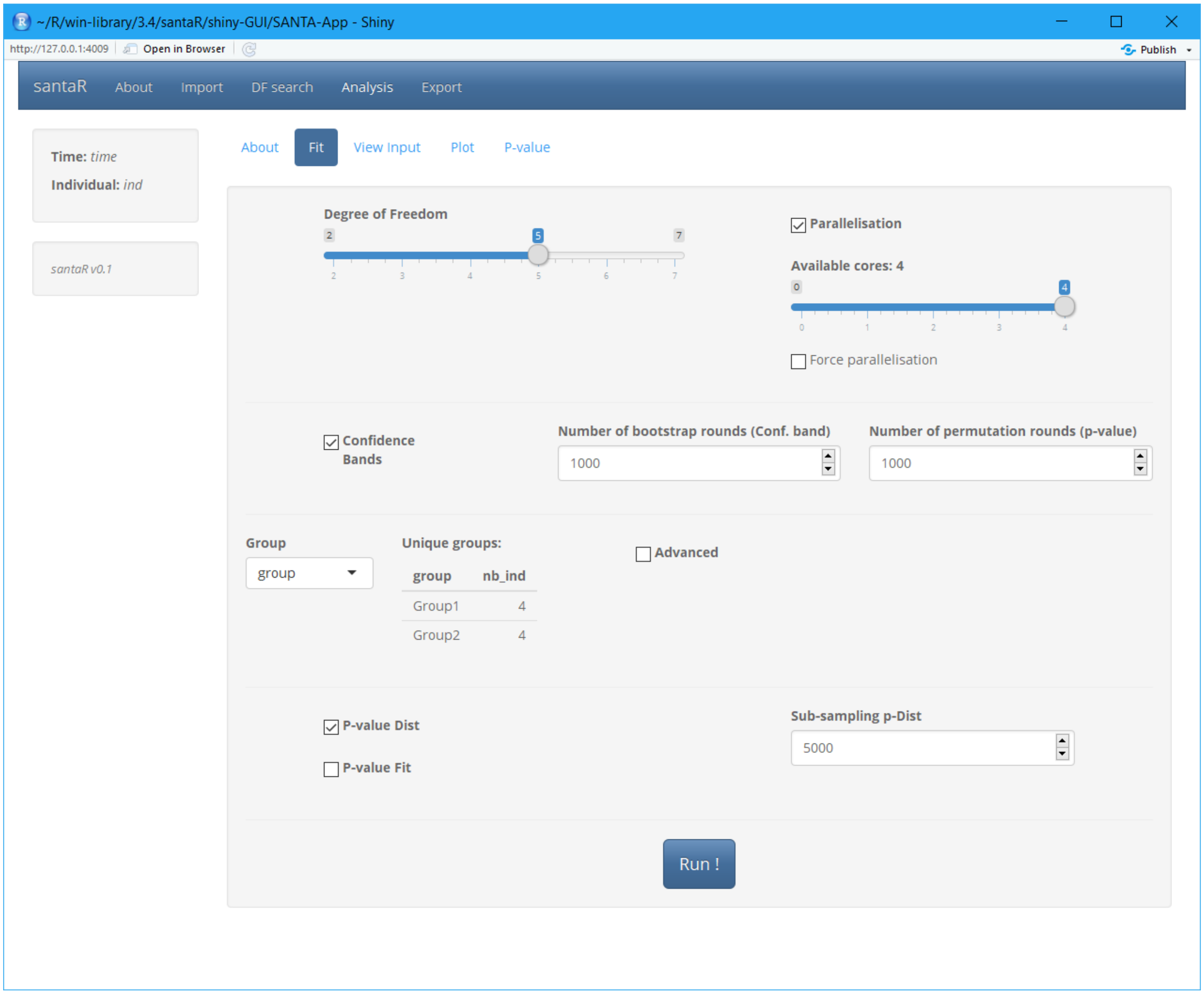
Open in Browser
Publish
santaR
About
Import
DF search
Analysis
Export
About
Auto-fit
Parameter Evolution
Plot fit
Missing value
Update
Run !
santaR v0.1
Fit PC 1 varExp=82.8% - df=5
Fit PC 2 varExp=6.8% - df=5
Fit PC 3 varExp=5.2% - df=5
Fit PC 4 varExp=2.9% - df=5
Fit PC 5 varExp=1.4% - df=5
Fit PC 6 varExp=0.8% - df=5
Projection
Time

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.



Time: *time*

Individual: *ind*

santaR v0.1

Degree of Freedom: 2 — 5 — 7

☑ Parallelisation

Available cores: 4

0 — 4

□ Force parallelisation

☑ Confidence Bands

Number of bootstrap rounds (Conf. band): 1000

Number of permutation rounds (p-value): 1000

Group: group

Unique groups:

group	nb_ind
Group1	4
Group2	4

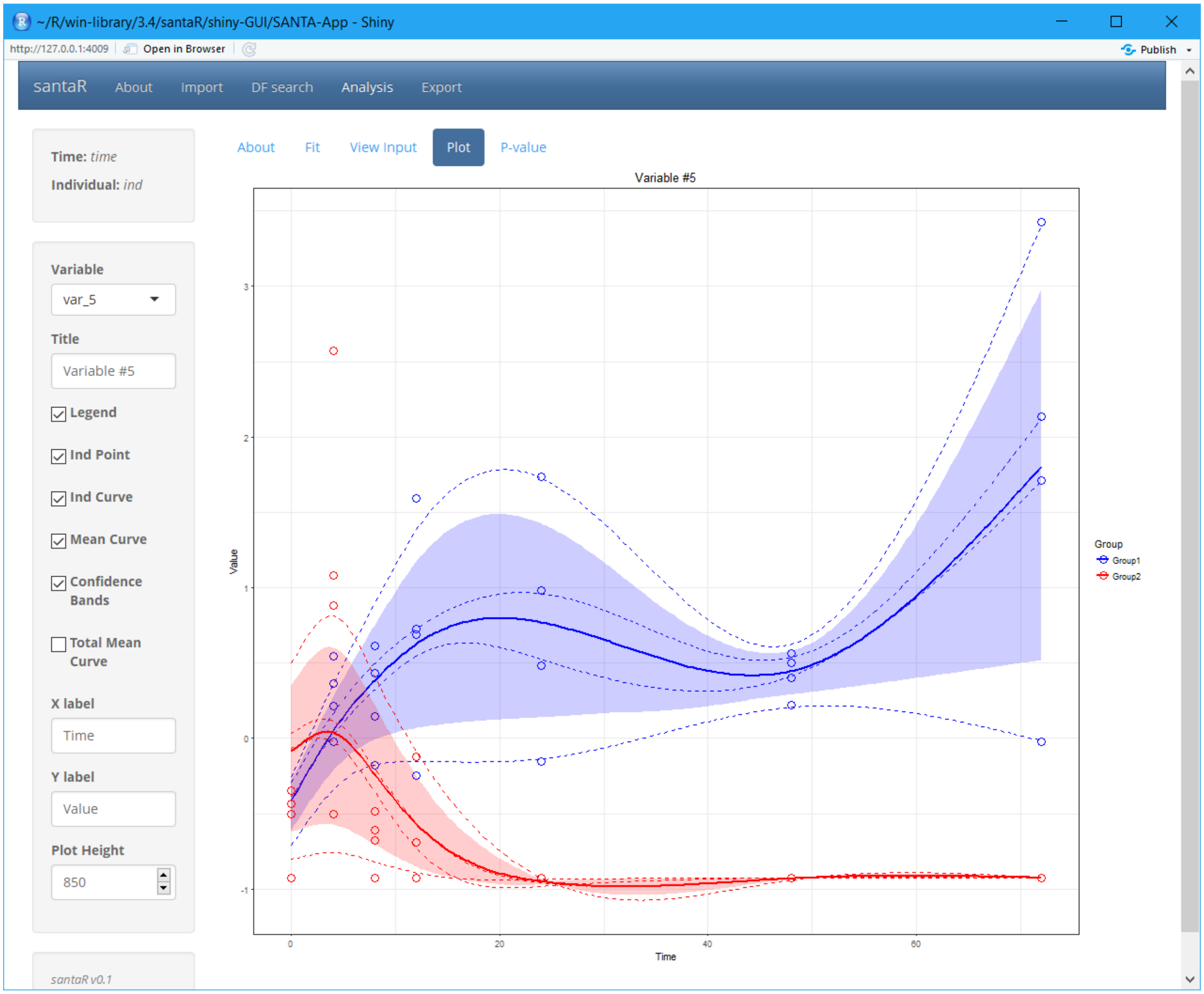
□ Advanced

☑ P-value Dist

□ P-value Fit

Sub-sampling p-Dist: 5000

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.



Time: *time*

Individual: *ind*

Variable
var_5

Title
Variable #5

☑ **Legend**

☑ **Ind Point**

☑ **Ind Curve**

☑ **Mean Curve**

☑ **Confidence Bands**

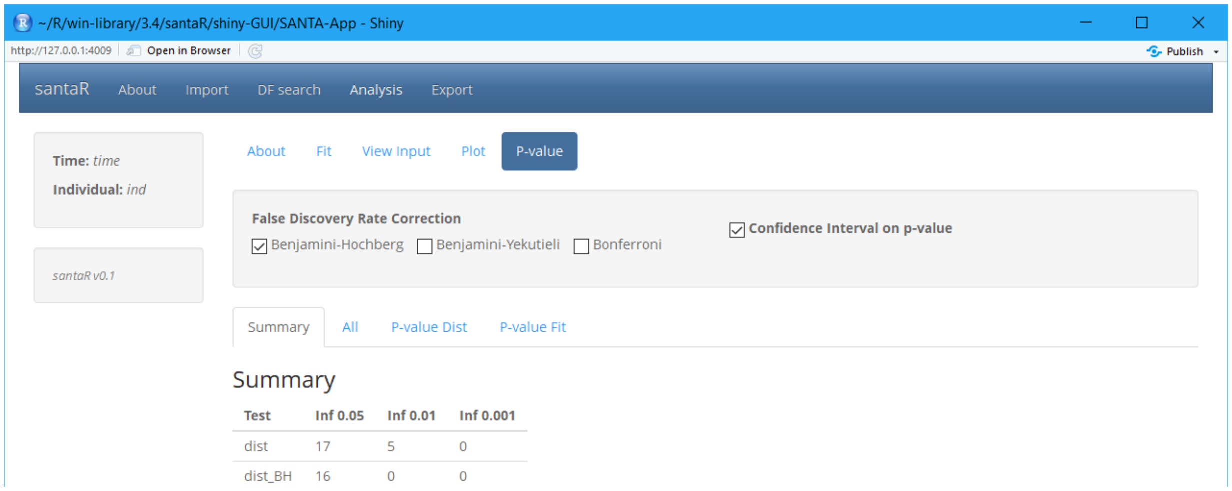
□ **Total Mean Curve**

X label
Time

Y label
Value

Plot Height
850

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.



Time: *time*

Individual: *ind*

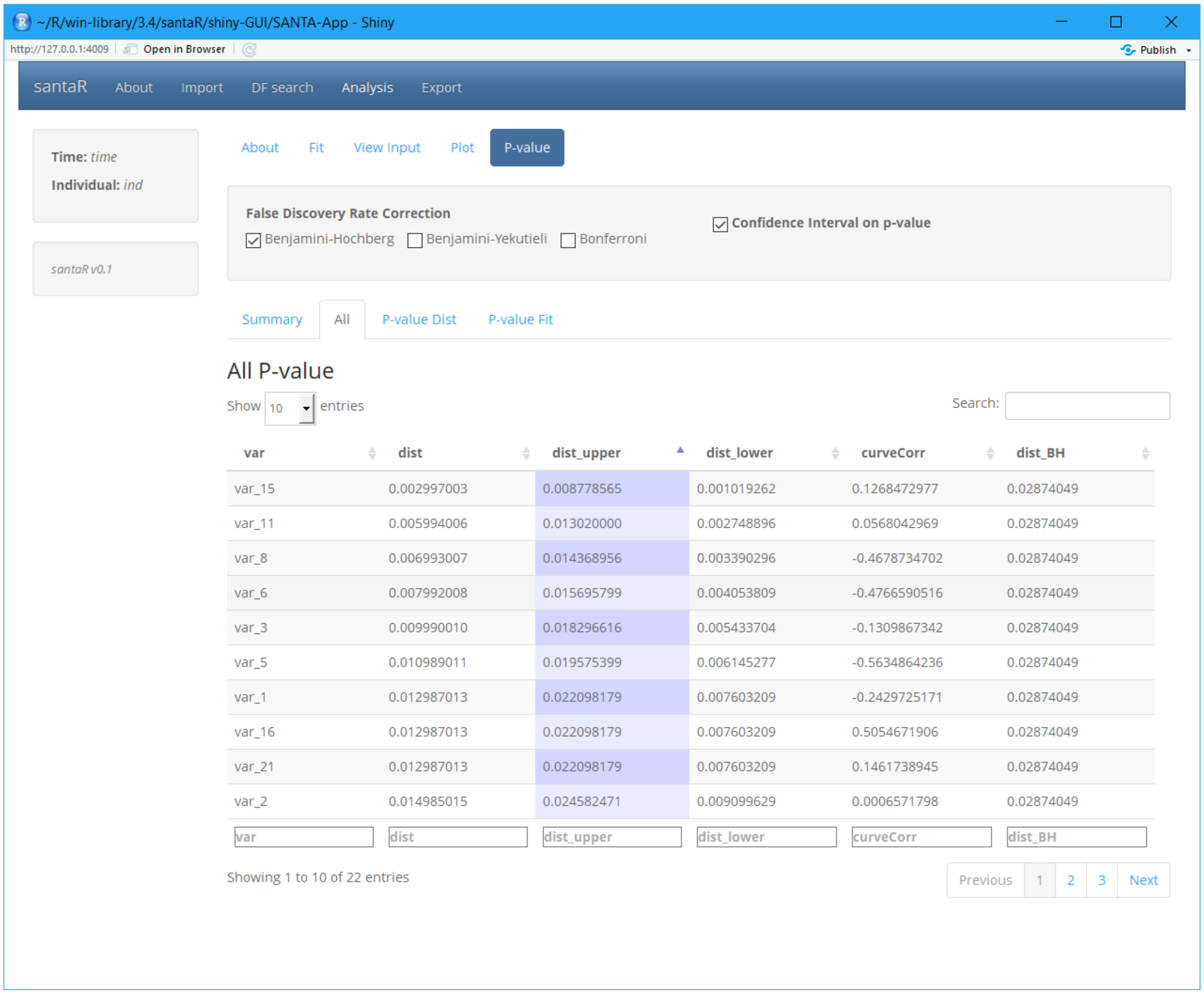
False Discovery Rate Correction

☑ Benjamini-Hochberg ☐ Benjamini-Yekutieli ☐ Bonferroni

☑ **Confidence Interval on p-value**

Summary

Test	Inf 0.05	Inf 0.01	Inf 0.001
dist	17	5	0
dist_BH	16	0	0



Time: *time*

Individual: *ind*

False Discovery Rate Correction

☑ Benjamini-Hochberg ☐ Benjamini-Yekutieli ☐ Bonferroni

☑ **Confidence Interval on p-value**

All P-value

Show 10 entries

Search:

var	dist	dist_upper	dist_lower	curveCorr	dist_BH
var_15	0.002997003	0.008778565	0.001019262	0.1268472977	0.02874049
var_11	0.005994006	0.013020000	0.002748896	0.0568042969	0.02874049
var_8	0.006993007	0.014368956	0.003390296	-0.4678734702	0.02874049
var_6	0.007992008	0.015695799	0.004053809	-0.4766590516	0.02874049
var_3	0.009990010	0.018296616	0.005433704	-0.1309867342	0.02874049
var_5	0.010989011	0.019575399	0.006145277	-0.5634864236	0.02874049
var_1	0.012987013	0.022098179	0.007603209	-0.2429725171	0.02874049
var_16	0.012987013	0.022098179	0.007603209	0.5054671906	0.02874049
var_21	0.012987013	0.022098179	0.007603209	0.1461738945	0.02874049
var_2	0.014985015	0.024582471	0.009099629	0.0006571798	0.02874049
var	dist	dist_upper	dist_lower	curveCorr	dist_BH

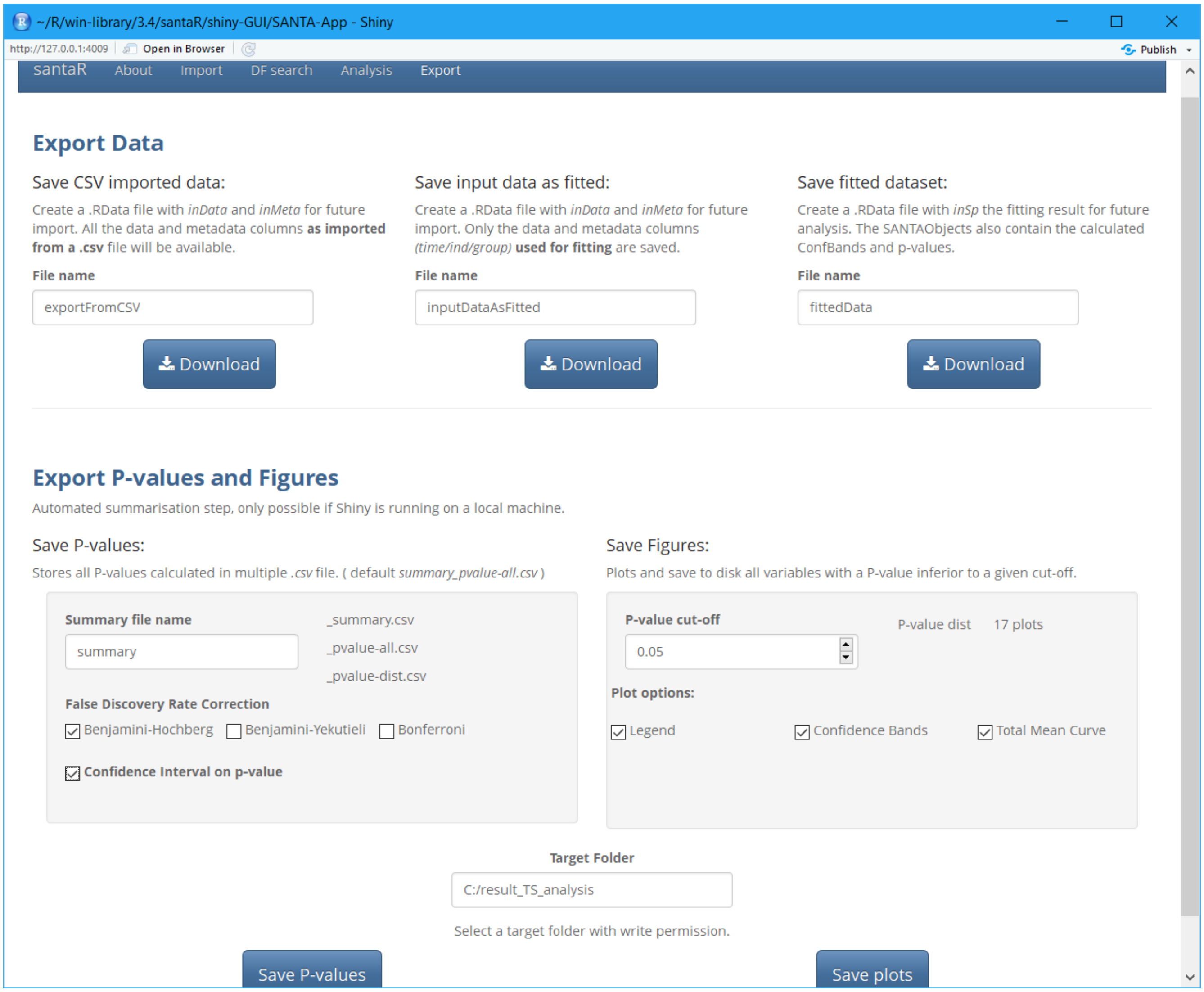
Showing 1 to 10 of 22 entries

Previous 1 2 3 Next

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.



Export Data

Save CSV imported data:

Create a .RData file with *inData* and *inMeta* for future import. All the data and metadata columns **as imported from a .csv** file will be available.

File name

exportFromCSV

Save input data as fitted:

Create a .RData file with *inData* and *inMeta* for future import. Only the data and metadata columns *(time/ind/group)* **used for fitting** are saved.

File name

inputDataAsFitted

Save fitted dataset:

Create a .RData file with *inSp* the fitting result for future analysis. The SANTAObjects also contain the calculated ConfBands and p-values.

File name

fittedData

Export P-values and Figures

Automated summarisation step, only possible if Shiny is running on a local machine.

Save P-values:

Stores all P-values calculated in multiple .*csv* file. (default *summary_pvalue-all.csv*)

Summary file name

summary

_summary.csv
_pvalue-all.csv
_pvalue-dist.csv

False Discovery Rate Correction

☑ Benjamini-Hochberg ☐ Benjamini-Yekutieli ☐ Bonferroni

☑ **Confidence Interval on p-value**

Save Figures:

Plots and save to disk all variables with a P-value inferior to a given cut-off.

P-value cut-off

0.05

P-value dist 17 plots

Plot options:

☑ Legend ☑ Confidence Bands ☑ Total Mean Curve

Target Folder

C:/result_TS_analysis

Select a target folder with write permission.

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