

User guide for the Auto-VTNA Python package and the Automatic VTNA Calculator GUI

Contents

1.	Preparing the kinetic data for Auto-VTNA.....	1
1.1.	Collecting kinetic data	1
1.2.	Importing the kinetic data	2
2.	User guide for the Auto-VTNA Python Code.....	4
2.1.	Visualising the kinetic data	4
2.2.	The VTNA selection dictionary	5
2.3.	Normal VTNA	6
2.4.	Automatic VTNA	7
2.5.	Automatic VTNA with fixed orders	8
3.	User guide for the Automatic VTNA Calculator GUI.....	9
3.1.	Uploading the Kinetic Dataset	9
3.2.	Inspecting the kinetic data	10
3.3.	Automatic VTNA Calculations	11
3.4.	Quick and Standard settings for the Automatic VTNA Calculation.....	12
3.5.	Saving the Results from an Automatic VTNA Calculation.....	12
3.6.	Cropping the Kinetic Data	12

1. Preparing the kinetic data for Auto-VTNA.

1.1. Collecting kinetic data

Auto-VTNA is a kinetic analysis technique for deriving reaction orders from concentration time data based on VTNA. Appropriate kinetic data can be collected by a range of different methods such as *ex situ* HPLC or GC, *in situ* IR, or by setting up reaction repeats and measuring the concentrations of products and/or reactants following workup at different times. For relevant examples, please see the literature examples referenced in Table 2. The datasets used to derive the results in Table 2 can also be accessed on the Auto-VTNA Github repository.²⁸

To derive the reaction order in a reaction species, its concentration profiles in at least two different excess experiments must be tabulated. If available, measured concentration values should be utilized.

Otherwise, the concentration profile of a reactant could be inferred from its known initial concentration and the measured concentration profile of the product or another reactant, assuming perfect mass balance. Additives and catalysts that are not consumed during the reaction are most often assumed to remain constant during the reaction. For examples of how VTNA datasets have been prepared in previous studies, see Table S7.

1.2. Importing the kinetic data

To utilize Auto-VTNA, the kinetic data should be organized into a .xlsx file in Excel with concentration time data from experiments separated into different sheets with appropriate names. Each sheet should contain the time axis in column A and the concentration profiles of different reaction species in the next columns (Fig. S1).

	A	B	C	D
1	Time /s	Product	Dimethyl itaconate	Piperidine
2	0	0	0.6	0.6
3	1140	0.22	0.39	0.4
4	2280	0.33	0.29	0.3
5	3600	0.38	0.23	0.22
6	5820	0.46	0.17	0.18
7	12420	0.52	0.09	0.11
8	16620	0.54	0.08	0.09

	A	B	C	D
1	Time /s	Product	Dimethyl itaconate	Piperidine
2	0	0	0.6	0.31
3	930	0.1	0.5	0.21
4	2010	0.16	0.44	0.15
5	3270	0.22	0.39	0.1
6	5490	0.26	0.34	0.05

	A	B	C	D
1	Time /s	Product	Dimethyl itaconate	Piperidine
2	0	0	0.6	0.45
3	1140	0.17	0.43	0.28
4	2400	0.24	0.35	0.18
5	4680	0.34	0.26	0.11
6	8100	0.39	0.21	0.06
7	11040	0.41	0.19	0.04

	A	B	C	D
1	Time /s	Product	Dimethyl itaconate	Piperidine
2	0	0	0.45	0.45
3	1680	0.18	0.29	0.27
4	3660	0.26	0.2	0.19
5	5340	0.29	0.16	0.15
6	8160	0.33	0.12	0.12

Figure S1: Screenshot of a representative .xlsx file which can be used as input for Auto-VTNA.

To avoid errors, ensure that the concentration profile of a given reaction species appears in the same column on each sheet of the .xlsx file. If the concentration profile for a reaction species is unknown or has not been measured for a given experiment, leave it empty. However, if this reaction species is selected as a normalized reaction species for a calculation involving the relevant experiment sheet, the calculation will fail and produce an error. It is generally recommended that users calculate all reaction species for each experiment, as this enables total VTNA calculations. Alternatively, if only the product and reactant with varied initial concentrations are known, you can prepare separate .xlsx files for each reaction species and perform automatic sequential VTNA on these.

To upload the .xlsx file to the Auto-VTNA Calculator, the user must click browse and select the relevant .xlsx file (see Section S3.1).

If Auto-VTNA is being used without its GUI, the .xlsx file should be imported using `pd.read_excel()`:

```
kinetic_data=pd.read_excel('filename.xlsx',sheet_name=None)
```

Setting `sheet_name` to “None” ensures that the `kinetic_data` variable is a dictionary with keys corresponding to the different experiments. The values of the experiment keys will be defined as pandas dataframe objects with the same Table structure as in the original .xlsx file.

It is important that there are no errors in the .xlsx file as this will cause Auto-VTNA to fail. For example, no non-numerical values should be in any sheet except for the column titles in row 1. Other requirements on the input data are as follows:

- The column headers in row 1 should be identical for each sheet.
- The time column needs to come first, i.e. in column A.
- The time column should be monotonically increasing.
- There should not be any empty sheets or sheets with values in incorrect position.

To aid this process, a function called “`check_kinetic_data`” has been written. The function checks that the kinetic data is loaded correctly by several criteria and generates a report which specifies any errors in the .xlsx file so help the user resolve the issue. This function should be applied first time a .xlsx file is being analysed by Auto-VTNA (Fig. S2).

```
Kinetic data check report.
-----
Check 1: Correct loading of kinetic data from Excel.
  ✓ The kinetic data is loaded as a dictionary.
  ✓ Each element associated a key in the kinetic data dictionary is a pandas dataframe object.
  ✓ The dataframe element(s) in the kinetic data dictionary all contain 4 columns.
Check 2: Column title consistency.
  ✓ All experiment datasets have the same column titles.
Check 3: Numerical datapoints.
  ✓ Every experiment dataset contains only numerical values.
Check 4: Positive datapoints.
  ✓ No datasets contain negative time or concentration values.
Check 5: Time column monotonicity.
  ✓ The time column (first column) of the dataset in every sheet is monotonically increasing.
Check 6: Time and concentration columns.
  ✓ All dataframe elements in the kinetic data dictionary has column titles which are strings.
  ✓ The column titles of every experiment dataset are all strings.
NB: Make sure that the time columns are the first columns in each dataframe in kinetic_data.
Currently, the first column titles are: Time /s.
Conclusion: The kinetic data has been loaded correctly if the time axis label is correct.
```

Figure S2: Report generated by `check_kinetic_data()` using a correctly loaded and formatted .xlsx file.

2. User guide for the Auto-VTNA Python Code

Users intending to utilize Auto-VTNA *via* its GUI (the Auto-VTNA Calculator) may skip to Section S3. In general, **the GUI is recommended** as it offers more features and is written for optimal user-friendliness.

2.1. Visualising the kinetic data

Once the kinetic data has been loaded successfully into Jupyter Notebook or a code editor like VSCode, it can be visualised in several ways using functions located in the `__init__.py` file of the package (accessed by typing `auto_vtna.function_name()`). For example, `plot_data()` generates graphs showing the concentration profiles in each experiment. The graphs can be generated side-by-side in one window by setting the `plot_mode` argument to 'together'. Alternatively, the graphs can be generated in an interactive manner where the user can use the right and left arrow keys to scroll through the experiments (`plot_mode='scrollable'`). Another function `plot_data_MB()` also exists which can be used to visualise the same data but with an added mass balance curve on a secondary axis specified by a list of reaction species names to include and another list of their corresponding stoichiometry (Fig. S3).

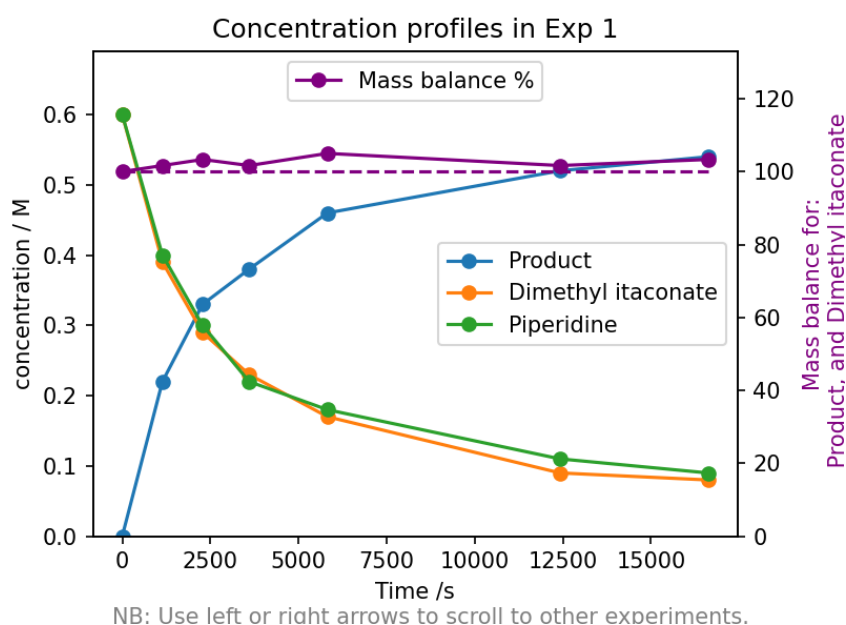


Figure S3: Graph visualising the product, dimethyl itaconate and piperidine concentration profiles (left axis) as well as the mass balance in dimethyl itaconate (right axis).

To visualize all the concentration profiles for a selected number of reaction species across all experiments, `plot_data_together()` can be used (Fig. S4). As well as the kinetic data variable, this function inputs a list of reaction species to be included (one species is recommended to avoid a cluttered plot). A function `initial_concs()` can be used to generate a Table containing the initial concentrations of every reaction species across all experiments. This can be useful to get an overview of the contents of each experiment dataset.

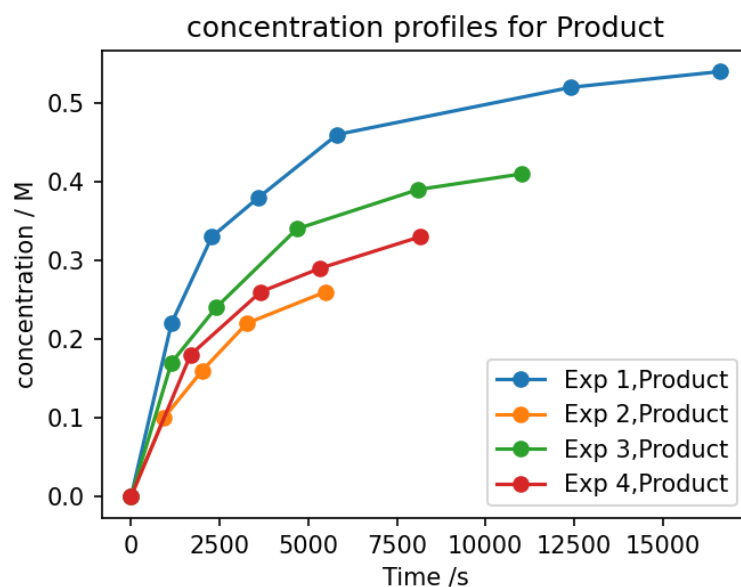


Figure S4: Concentration profiles of the product across all experiments generated using `plot_data_together()`.

Each of these data visualisation functions discussed above contain a range of keyword arguments to specify units, figure size, number of decimal places, titles etc. and the user is advised to consult the `__init__.py` file of the package for more information.

2.2. The VTNA selection dictionary

To perform automatic or normal VTNA, a VTNA selection dictionary is needed. The selection dictionary contains information necessary to carry out variable time normalisation such as which experimental datasets to analyse ('RM' for reaction mixture) and whether some datapoints or datapoint ranges should be omitted from the analysis. The omissions can either be provided as a list of datapoints or as a range of values where 100 is defined as the highest time ('range' followed by two numbers such as 50 and 100 to remove the second half of the datapoints). The selection dictionary also specifies which reaction species to use for normalising the time scale ('normalised species') with their respective reaction orders. The output species of the calculation is also specified ('output species'). A selection dictionary template can be generated using the function `make_VTNA_selection()` for the given dataset (Fig. S5A) which can be completed by specifying the experiments, any datapoint or datapoint range omissions, the output species and the normalised species, with their respective order values (Fig. S5B).

A:`auto_vtna.make_VTNA_selection(data)`

```
{'RM': {'Exp 1': {'omissions': None},
'Exp 2': {'omissions': None},
'Exp 3': {'omissions': None},
'Exp 4': {'omissions': None}},
'output_species': 'OS',
'normalised_species': {'NS': 1}}
```

B:`selection`

```
{'RM': {'Exp 1': {'omissions': [4]},
'Exp 2': {'omissions': None},
'Exp 3': {'omissions': None}},
'output_species': 'Product',
'normalised_species': {'Dimethyl itaconate': 1,
'Piperidine': 1}}
```

Figure S5: A: VTNA selection dictionary for the example dataset in Figure S9 generated automatically using the function `make_VTNA_selection()`. B: Properly filled out VTNA selection dictionary for the same dataset with experiments, output species and normalised species specified.

2.3. Normal VTNA

While the Auto-VTNA functions are readily available in the `__init__.py` file of the package, accessible through importing `auto_vtna` and using `auto_vtna.function_name()`, the conventional VTNA code is structured as a class within a distinct Python file named `Normal_VTNA.py`. By conventional VTNA, it is meant that the output of `Normal_VTNA` contains the normalised time axis ('tT') of each experiment calculated using the order values and normalised species given by the VTNA selection dictionary (Fig. S6).

```
from auto_vtna.Normal_VTNA import Normal_VTNA
VTNA_etoh=Normal_VTNA(data,selection)
VTNA_etoh.resultt
```

Exp 1:

	Time /s	Product	Dimethyl itaconate	Piperidine	tT
0	0	0.00	0.60	0.60	0.000
1	1140	0.22	0.39	0.40	282.150
2	2280	0.33	0.29	0.30	417.810
3	3600	0.38	0.23	0.22	507.042
4	12420	0.52	0.09	0.11	739.890
5	16620	0.54	0.08	0.09	775.590

Exp 2:

	Time /s	Product	Dimethyl itaconate	Piperidine	tT
0	0	0.00	0.60	0.31	0.0000
1	930	0.10	0.50	0.21	132.9900
2	2010	0.16	0.44	0.15	224.3580
3	3270	0.22	0.39	0.10	289.7205
4	5490	0.26	0.34	0.05	350.4930

Exp 3:

	Time /s	Product	Dimethyl itaconate	Piperidine	tT
0	0	0.00	0.60	0.45	0.0000
1	1140	0.17	0.43	0.28	214.2915
2	2400	0.24	0.35	0.18	327.3135
3	4680	0.34	0.26	0.11	428.1465
4	8100	0.39	0.21	0.06	496.4610
5	11040	0.41	0.19	0.04	525.8610

Figure S6: The output of a Normal VTNA calculation is found in the rightmost column 'tT' which can be used to generate a VTNA overlay plot (see below).

Once the instant Normal VTNA calculation has been run, the result can be visualised by calling the method `.plot_VTNA()` to generate a conventional VTNA overlay plot. 13 keyword arguments can be defined to customize the plot such as "size_scaler", "linewidth", "xtick_rotation" etc. (see documentation for more details). Alternatively, the `plot_VTNA_with_overlay_score()` method can be called to generate the overlay plot together with a line of best fit, either an ordinary or monotonic polynomial of a selected degree, or a straight line through origin. The selected overlay score, set to RMSE by default, will then be shown in the graph title (Fig. S7). If a degree of 1 has been defined, the title will also contain the slope of the fitted line with its standard error uncertainty. This slope can be useful for determining the observed rate constant when linearisation has occurred (Fig. S7C)

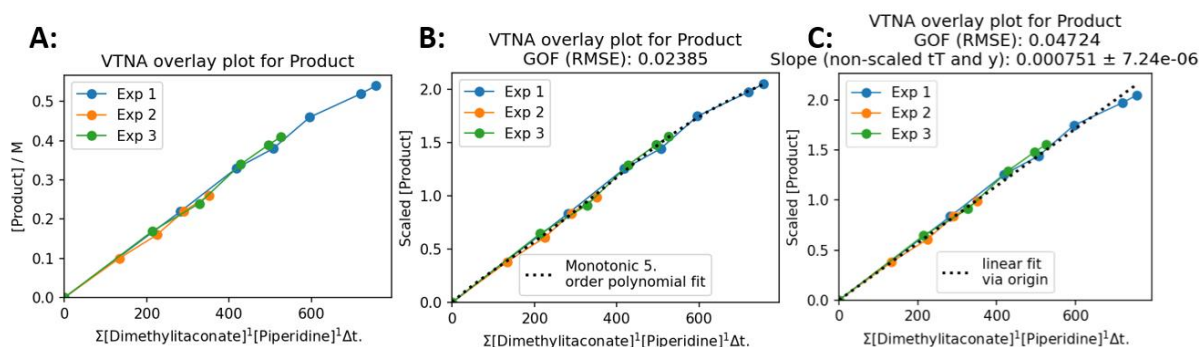


Figure S1: VTNA overlay plots generated by calling `plot_VTNA()` (A) or `plot_VTNA_with_overlay_score()` with `deg=5` and `constraint='monotonic'` (B) or `deg=1` and `constraint='through origin'` (C).

2.4. Automatic VTNA

As for Normal_VTNA, the automatic VTNA code is structured as a class within a distinct Python file in the Auto_VTNA package named `Automatic_VTNA.py`. An instance of Automatic VTNA is initiated by inputting the kinetic data and the VTNA selection dictionary to define the output reaction species and the normalised reaction species for which the optimal order values will be determined.

Nine keyword arguments can be specified to modify the settings of the calculation such as the number of iterations, the range of order values to investigate and the resolution of the order value mesh (see documentation for more details). The calculation will then run, and the best order values of each iteration printed in the terminal. Once the calculation is complete, the results can be viewed via the `result`, `best_orders`, `interval`, and `best_slope` attributes of the `Automatic_VTNA` object:

- The `result` attribute contains the overlay scores obtained across all reaction order combinations.
- The `interval` attribute contains the 15% uncertainty intervals for each normalised reaction species as discussed in the main publication.

If the calculation was set to elucidate the order values in 1 or 2 reaction species, the results can be visualized by calling the `plot_orders_vs_overlay()` method of the `Automatic_VTNA` class (see Fig. 3 and 7).

To customise the Figures, 16 different keyword arguments such as “`zoom`”, “`y_unit`” and “`interval`” can be adjusted. The graph or contour plots generated by `plot_orders_vs_overlay()` are interactive and will generate the overlay plot corresponding to the order values of the click point, either with (right-click) or without (left-click) the overlay score and fitted polynomial (see Fig. 4).

2.5. Automatic VTNA with fixed orders

For an automatic VTNA calculation, any number of normalised reaction species can be selected in the VTNA selection dictionary as long as its name corresponds to a column in the uploaded kinetic dataset (dictionary of Pandas dataframe objects). However, a list of “fixed order species” can also be provided which will be fixed at the order value specified in the VTNA selection dictionary. Instead of trialling a range of order values for the fixed reaction species, the time axis will remain normalised with these reaction species at the specified order value, and rather optimise the order values of the non-fixed normalised species. As mentioned in the main publication, this effectively reduces the dimensionality of the automatic VTNA calculation.

As an example, the correlation between overlay score and orders in reactants **14** and **15** for the entire kinetic data on the reaction system in Entry 3 in Table S2 can be visualised by fixing the catalyst order to 1 (Fig. S8A). Likewise, the correlation between overlay score and orders in reactants **18** and **19** for the entire kinetic data on the reaction system in Entry 4 in Table S2 can be visualised by fixing the catalyst and base orders to 1 and 0 respectively (Fig. S8B).

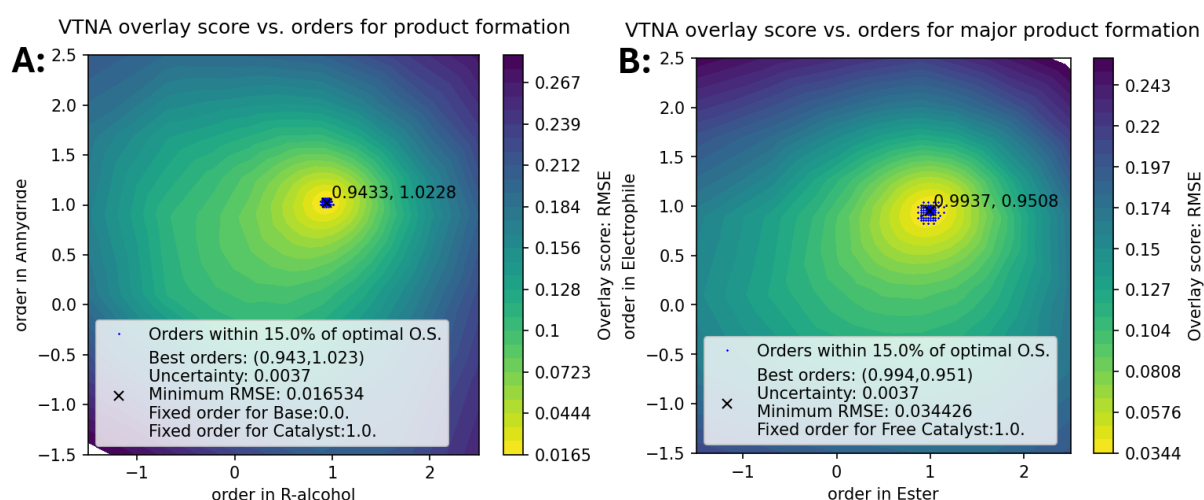


Figure S8: Overlay score vs. reactant orders for the kinetic data on the reaction systems in Entries 3 (A) and 4 (B) in Table S2, obtained via automatic VTNA calculations on the entire dataset in which the catalyst orders were set to the previously calculated value of 1 and the base order fixed at 0 to generate contour plot A. See additional information in the figure captions.

3. User guide for the Automatic VTNA Calculator GUI

A graphical user interface (GUI) named the “Automatic VTNA Calculator” has been developed to streamline the use of the features of the Auto-VTNA Python package without the need for coding knowledge.

The following sections outline the recommended steps for carrying out automatic VTNA using the Auto-VTNA Calculator. The workflow can be summarized into four essential steps:

1. Upload a correctly formatted .xlsx file.
2. Select experiments, normalised reaction species for which orders should be derived and an output reaction species. (Note, initially the “quick setting” is best used if more than 4 reaction species are investigated).
3. Click run under “Automatic VTNA” to calculate the order value which optimises the overlay score for each normalised species.
4. Visualise the results of the calculations via “VTNA Overlay Plots” or by clicking “Plot Results” if 1 or 2 normalised reaction species have been selected.

A YouTube tutorial has also been produced running through these steps, which can be found linked to the GitHub: <https://github.com/ddalland/Auto-VTNA>.

The following sections go into more details of each feature of the Automatic VTNA calculator.

NOTE: Be aware that the results from an automatic VTNA calculation are only meaningful if the following criteria have been met:

- The different sheets in the .xlsx file corresponds to experiments carried out with identical reaction conditions apart from varying initial concentrations of reactants, additives or catalysts.
- To obtain a meaningful order value in a selected normalised reaction species, experiments with different initial concentrations in the reaction species must be selected.
- The kinetic data must be of sufficiently high data point density for accurate numerical integration (at least 5 data points per concentration profile is recommended).
- An appropriate total fitting method must be selected.

3.1. Uploading the Kinetic Dataset

Once the Auto-VTNA Calculator executable file has been run, the dashboard of the Auto-VTNA Calculator is activated by uploading the kinetic data as an .xlsx file (browse button) or as CSV files (Use

CSVs). This automatically populates the “Select Experiments”, “Select Normalised Species” and “Select Output Species” boxes (Fig. S9).

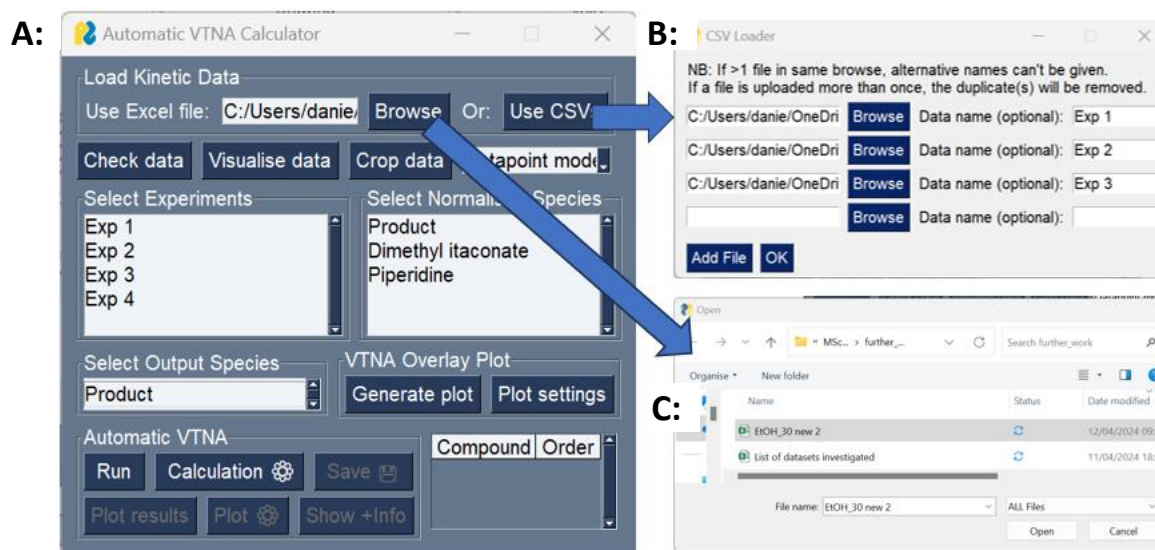


Figure S9: Uploading kinetic data onto the Automatic VTNA dashboard (A). This can be done either by selecting the relevant .xlsx file (C) or by clicking use CSVs and selecting the desired number of CSV files (C).

3.2. Inspecting the kinetic data

Once the kinetic data has been uploaded, it can be checked by the “Check Data” button or visualised in different ways by clicking “Visualise Data”. For example, an initial concentration Table can be generated by clicking “Visualise Data” followed by “Generate Initial Concentration Table” (Figure S10). The kinetic data can also be visualised in different ways to generate plots as shown in Section 2.1.

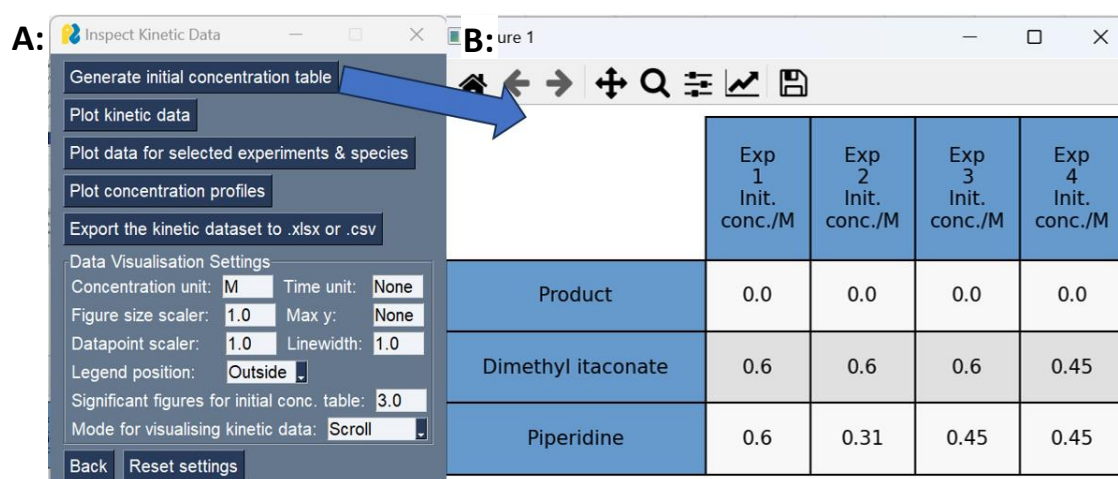


Figure S10: The “Inspect Kinetic Data” menu generated by clicking Visualise Data (A). From this sub-window, several different Auto-VTNA functions can be accessed, such as generation of an initial concentration Table (B).

Several settings can also be adjusted such as the position of legends, the size of Figures and Concentration units, providing more user-friendly access to the keyword arguments of the underlying Auto-VTNA functions.

3.3. Automatic VTNA Calculations

To perform automatic VTNA calculations or generate VTNA overlay plots, the user must select:

- the normalised reaction species;
- the experiment datasets to include; and
- the output reaction species (normally the product).

Once these inputs have been selected, ordinary VTNA overlay plots can be generated by clicking “Generate Plot” within the “VTNA Overlay plot” frame (Fig. S11). If desired, normal VTNA can be performed by then manually altering order values for the reaction species.

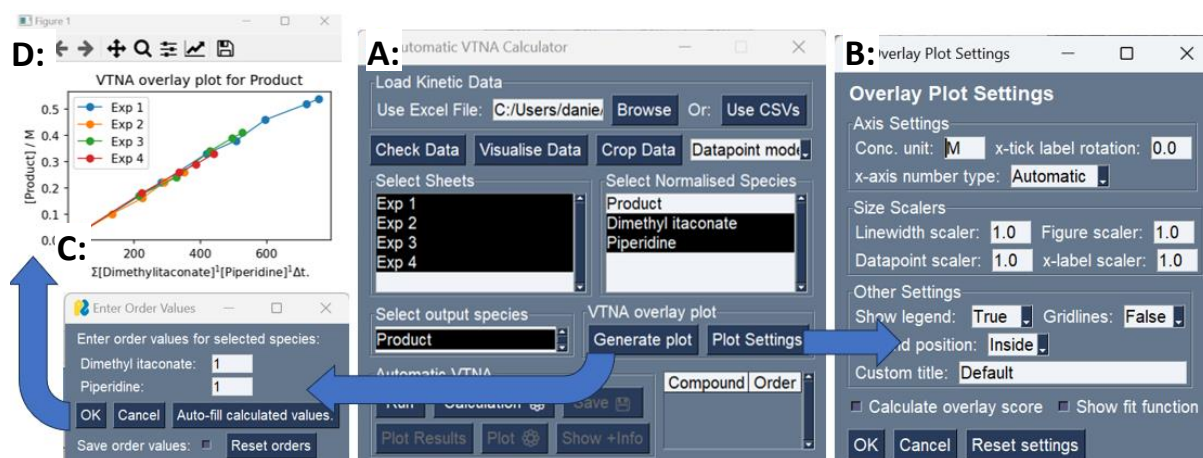


Figure S11: Illustration of how the Automatic VTNA Calculator (A) can be used to generate overlay plots (D) by specifying the relevant order values (C). The VTNA overlay plots can be modified via the “Plot Settings” menu (B).

The reaction order values of the “Selected Normalised Species” that maximise concentration profile overlay can be calculated automatically by clicking “Run” in the “Automatic VTNA” frame. Once the calculation has been completed, the results can be visualised by clicking plot results. This will generate an overlay score vs. order value graph if one normalised species has been defined, or a contour plot if two normalised species have been defined.

The Automatic VTNA settings can be accessed and modified by clicking on the calculation and plot setting buttons. The calculated orders will also appear in the bottom left Table. Moreover, the order value intervals and overlay score of the best order values can also be shown by clicking on the “Show + Info” button (Fig. S12). The percentage cut-off defining the order value intervals can also be updated to generate a new table.

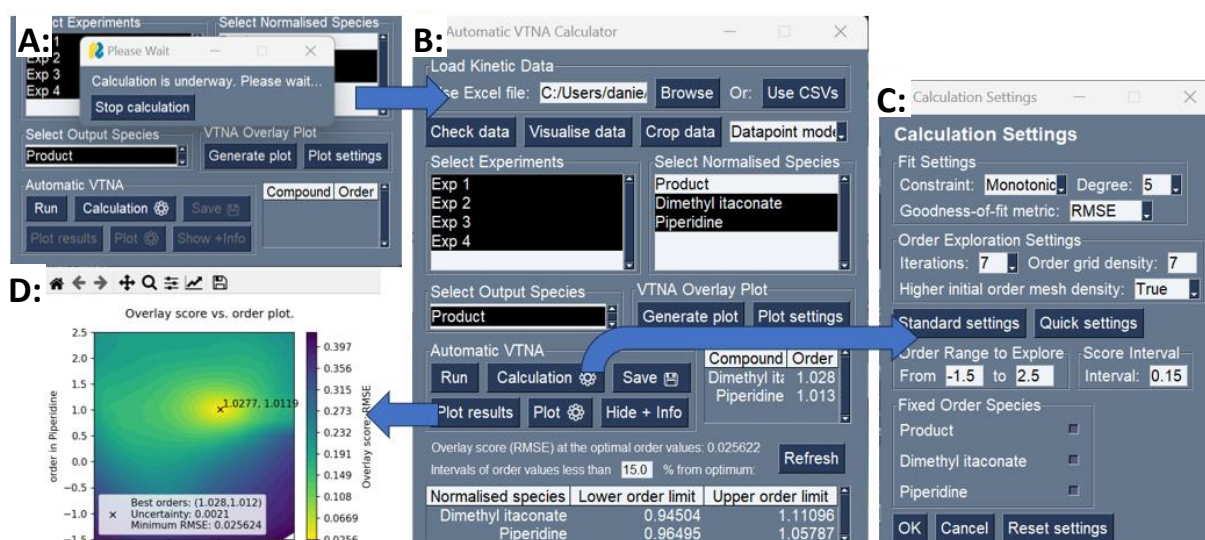


Figure S12: After running an automatic VTNA calculation with two normalised reaction species (A), the results can be visualised by clicking “Plot Results” to generate an overlay score versus order contour plot (D). The calculation and plot settings buttons can be used to alter the standard keyword arguments of the `Automatic_VTNA` class and its plotting methods (C).

3.4. Quick and Standard settings for the Automatic VTNA Calculation

The time taken for an Automatic VTNA calculation depends on several settings such as the number of algorithm iterations, the resolution of the order value grid of each resolution (section S7 in the ESI of the Auto-VTNA publication). **To minimise the calculation time**, a “Quick Settings” button has been included in the calculation settings menu which applies iteration and order grid density parameters that optimise the processing time. Alternatively, the “Standard Settings” box can be clicked to apply settings that are slower but cover a denser range of order value combinations in each iteration. The settings menu can also be used to specify the range of order values to investigate, and fixed order values for any of the normalised reaction species to lower the dimensionality of the calculation as discussed in Section S7 in the ESI of the Auto-VTNA publication.

3.5. Saving the Results from an Automatic VTNA Calculation

To save the results from an automatic VTNA calculation using the Automatic VTNA Calculator, the button “Save” can be clicked. The resulting .xlsx or .csv file will contain the applied automatic VTNA settings, the overlay score of every trialled order value combination as well as the uncertainty intervals. If the results are saved as a .xlsx file, the most recently generated overlay vs. order plot will also be included.

3.6. Cropping the Kinetic Data

If necessary, parts of the loaded kinetic data can be removed before performing automatic VTNA by clicking the “Crop” button. If all experiments contain less than 25 time points, the default cropping setting is set to “Datapoint mode” in which the datapoints of each experiment are numbered and can be ticked for removal (Fig. S13B). Alternatively, “Range mode” can be selected, giving rise to a menu

in which ranges of time values (either absolute values or as a percentage of t_{end}) can be selected for each experiment within which all datapoints are removed (Fig. S13C). Range mode also allows the datapoint density of experiments to be reduced by only keeping every X datapoint, useful for highly dense datasets which give rise to slower automatic VTNA calculations. The cropping applied by these methods is reversible and can be removed by clicking “Reset” followed by “OK”.

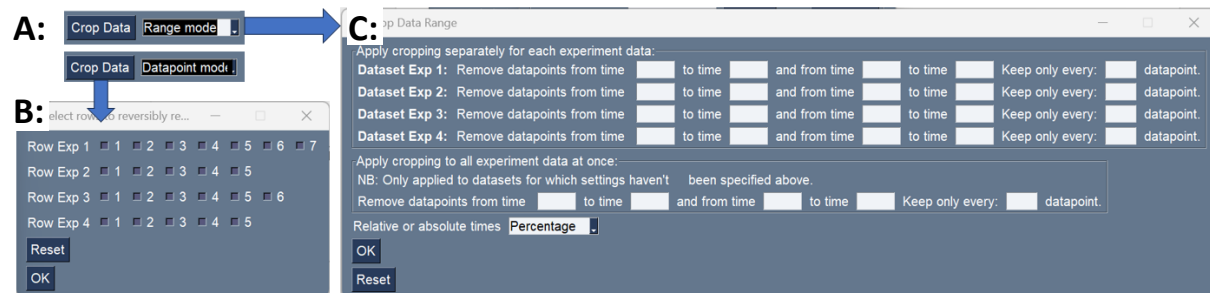


Figure S13: Data cropping menus for “Datapoint mode” (B) and “Range mode” (C) for modifying the uploaded kinetic data. The relevant menu can be generating by clicking “crop” once the correct data cropping mode has been selected.