

SOFTWARE METAPAPER

vaCATE: A Platform for Automating Data Output from Compartmental Analysis by Tracer Efflux

Rubens Flam-Shepherd¹, Dev T. Britto¹ and Herbert J. Kronzucker²

¹ Department of Cell and Systems Biology, University of Toronto, CA

² Faculty of Veterinary and Agricultural Sciences, University of Melbourne, AU

Corresponding author: Rubens Flam-Shepherd (ruben.flam.shepherd@mail.utoronto.ca)

Compartmental analysis by tracer efflux (CATE) is fundamental to examinations of membrane transport, allowing study of solute movement among subcellular compartments with high temporal, spatial, and chemical resolution. CATE can provide a wealth of information about fluxes and pool sizes in complex systems, but is a mathematically intensive procedure, and there is a need for software designed to fully, easily, and dynamically analyse results from CATE experiments. Here we present vaCATE (Visualized Automation of Compartmental Analysis by Tracer Efflux), a software package that meets these criteria. A robust suite of test cases using CATE datasets from experiments with intact rice (*Oryza sativa* L.) root systems reveals the high fidelity of vaCATE and the ease with which parameters can be extracted, using a three-compartment model and a curve-stripping procedure to distinguish them on the basis of variable exchange rates. vaCATE was developed using Python 2.7 and can be used in most situations where compartmental analysis is required.

Keywords: Compartmental analysis; tracer efflux; CATE; Python; data visualization; software; plant physiology; membrane transport; vaCATE

Funding Statement: This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Ontario Graduate Scholarship Fund (OGS).

(1) Overview

Introduction

Compartmental analysis is a methodology used to investigate the fluxes and accumulation of matter and energy in compartmented systems [1]. It is used to model phenomena from a diverse breadth of areas, including pharmacokinetics [2, 3], geography [4], ecology [5, 6], oncology [7, 8], physiology [9, 10, 11, 12], and even building design [13]. This broad utility is facilitated by the uniform theory compartmental analysis is built upon, and the systematic modelling methods it provides [1].

In plant physiology, compartmental analysis by tracer efflux (CATE) is one of several radioisotope methods used to trace the movement of substances across plant cell membranes and between plant organs. However, CATE can be used to do this non-invasively, which is unusual due to the structural and functional complexity of plant systems [14, 15]. In addition, although mathematically demanding and labour-intensive compared to other radioisotope methodologies, CATE can provide a more comprehensive view of the multiplex of unidirectional fluxes and compartmented concentrations of the traced substance [16]. Currently, a three-compartment model consisting of surface film, cell wall, and cytosol (in order of

decreasing rapidity of exchange) is generally accepted for use with CATE under short-term labelling scenarios [17, 18, 19, 20, 21, 22, 23], although it may not be valid under certain experimental conditions (e.g., high substrate concentrations, at which transport is governed by low-affinity systems) [24, 25, 26].

CATE involves measuring the kinetics of tracer release from labelled plant systems to the external solution. Typically, the procedure involves first “loading” plants with an appropriate radioisotope, by immersing roots of intact plants in radioactive solution long enough to achieve significant accumulation of radioisotope in the cytosol of the root cells, but short enough that potentially confounding fluxes from the vacuole remain unlabelled [21, 27, 28, 29]. After this, labelled plants are transferred to “efflux funnels” [20, 30] where they are “washed”, or eluted of radioactivity by exposure to sequential aliquots of non-radioactive nutrient solution. In each aliquot, solution exchange with labelled roots takes place over a prescribed time interval, after which the aliquot is removed (via a drainage clip attached to the efflux funnel), and immediately replaced with the next aliquot of non-radioactive solution. After the elution series is complete, plant organs are harvested and radioactivity

in plant organs and eluates is measured. The pattern of tracer release from the three-compartment system takes the form of a compound exponential [23] consisting of three release phases/compartments, and can be modelled by the equation

$$A_t = A_1 e^{-k_1 t} + A_2 e^{-k_2 t} + A_3 e^{-k_3 t},$$

In which A_t is the rate of total tracer release, A_n is the maximal tracer released from each compartment, and k_n is the rate constant for each compartment (i.e., the linearized slopes of declining tracer release found using semi-logarithmic transformation of data; see **Figure 4**). For a visual representation of a typical CATE procedure, see Coskun et al. (2014) [31].

To enable the rapid extraction of CATE parameters, vaCATE was developed to extend the functionality of an internal laboratory tool that consisted of a Visual Basic macro embedded in an Excel file. It improves upon the macro in several key areas: improved statistical analysis, the inclusion of a curve-peeling procedure that allows more rapidly exchanging compartments to be analysed, and error checking and handling. While there are a variety of software applications available for compartmental analysis (e.g., SAAM II [32], PKSolver [33], TopFit [34], to list a few), most are commercially-available and/or specialized for pharmacokinetics. In contrast, vaCATE is available for free, was developed specifically for use with CATE, and its open-source licensing agreement allows for modification of the underlying software architecture if non-commercial use in other applications is desired.

Implementation and architecture

Written in Python, vaCATE was constructed to allow users to easily manipulate and extract CATE parameters from large datasets. This is done through a front-end graphical user interface (GUI) implemented using wxPython and matplotlib, and a back-end implemented using xlswriter, xlrd, and numpy.

Initial Set-Up

After opening vaCATE, users are greeted by a menu dialog asking them to either analyse CATE data or generate a CATE template file (**Figure 1**). Users must format their data using a generated CATE template file before they are able to conduct an analysis using vaCATE. An example of a completed template file is given in **Figure 2** (and in the 'Examples' folder where vaCATE was cloned/installed). In this file, each individual CATE replicate is allocated its own column (from column 'C' onwards) with specific parameters input according to row labels in columns 'A' and 'B'. All of the information indicated by the labels in rows '1' through '7' is required. Of note, row 6 ('G-factor') allows users to apply instrument-specific corrections to account for inaccuracies in radioactivity measurements due to internal geometries of the detecting equipment [1, 35].

Under cell 'B8' (**Figure 2**), the time at which each eluate was removed is entered in descending order (from the first one taken onwards). It should be noted that eluate times are represented as decimal fractions of a minute

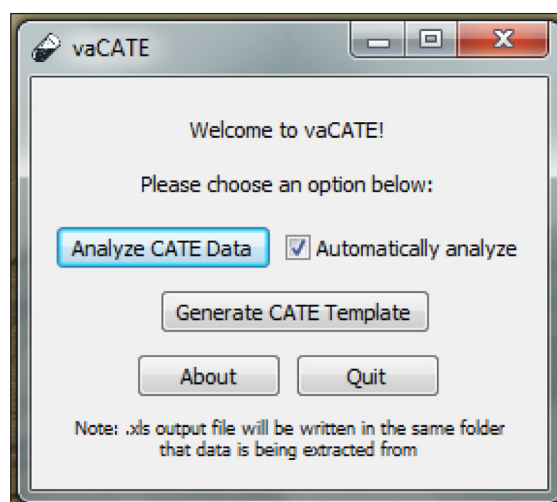


Figure 1: Initial menu dialog presented to the user.

	A	B	C	D
1	Run Name		Run 1	Run 2
2	Specific Activity (cpm · μmol⁻¹)		8593.046	8701.939
3	Root Cnts (cpm)		4469.1	5264.8
4	Shoot Cnts (cpm)		4133.7	3296
5	Root weight (g)		0.2246	0.2808
6	G-Factor		1.030861	1.023255
7	Load Time (min)		60.00	60.00
8	Vial #	Elution time (min)		
9	1	1.5	94470.7	150324.8
10	2	3	20995.9	28904.4
11	3	4.5	6732	9978
12	4	6	4048.2	5365
13	5	7.5	2626.3	3250.6
14	6	9	1794.2	2480.6
15	7	10.5	1499.4	1775.4
16

Figure 2: Example of a completed CATE template file. Cells containing '...' are used to indicate that the spreadsheet contains data below what is presented in the figure. Note that completion of the 'Vial #' column under cell 'A8' is optional.

(e.g., 1 minute and 30 seconds is represented as 1.5 minutes). This sequence of elution times is used for every CATE replicate in the file. If different elution schedules are used in the same dataset then a separate CATE template file should be prepared for each schedule. Within the columns allocated to each CATE replicate, eluate radioactivities are entered next to their corresponding elution times (found under cell 'B8'; **Figure 2**). In this way, data from CATE replicates sharing the same elution schedule are processed simultaneously.

Once the user's dataset has been entered into the CATE template file, the data can be analysed by clicking on the

'Analyze CATE Data' button (**Figure 1**). Users have the option to 'Automatically Analyze' the data in the CATE template file (default option; **Figure 1**), which conducts an objective regression using the last 8 eluates (see Types of Regression Models section below). If this feature is disabled by deselecting the corresponding checkbox then no analysis is conducted when the initial dynamic preview of the data is presented (**Figure 4**).

Experiment and Analysis Object Instantiation

Once a correctly filled-out CATE template file is selected for analysis, `Excel.grab_data()` retrieves the data from the template file and returns an Experiment object (**Figure 3**). The 'analyses' attribute of this Experiment

object (`Experiment.analyses`) contains a list of Analysis objects; one Analysis object is created for each CATE replicate found in the template file. In turn, the 'run' attribute of each Analysis object (`Analysis.run`) points to a Run object which contains the immutable data for that CATE replicate. This is the data that does not depend on the **type** of analysis that will eventually be done (e.g., the shoot/root weight, the elution times, etc.; see **Figure 3** for complete set of parameters).

Dynamic Preview of Data

Experiment objects are then loaded into a wx. Frame object for display in the graphical user interface (**Figure 4**). The name of the current CATE replicate being

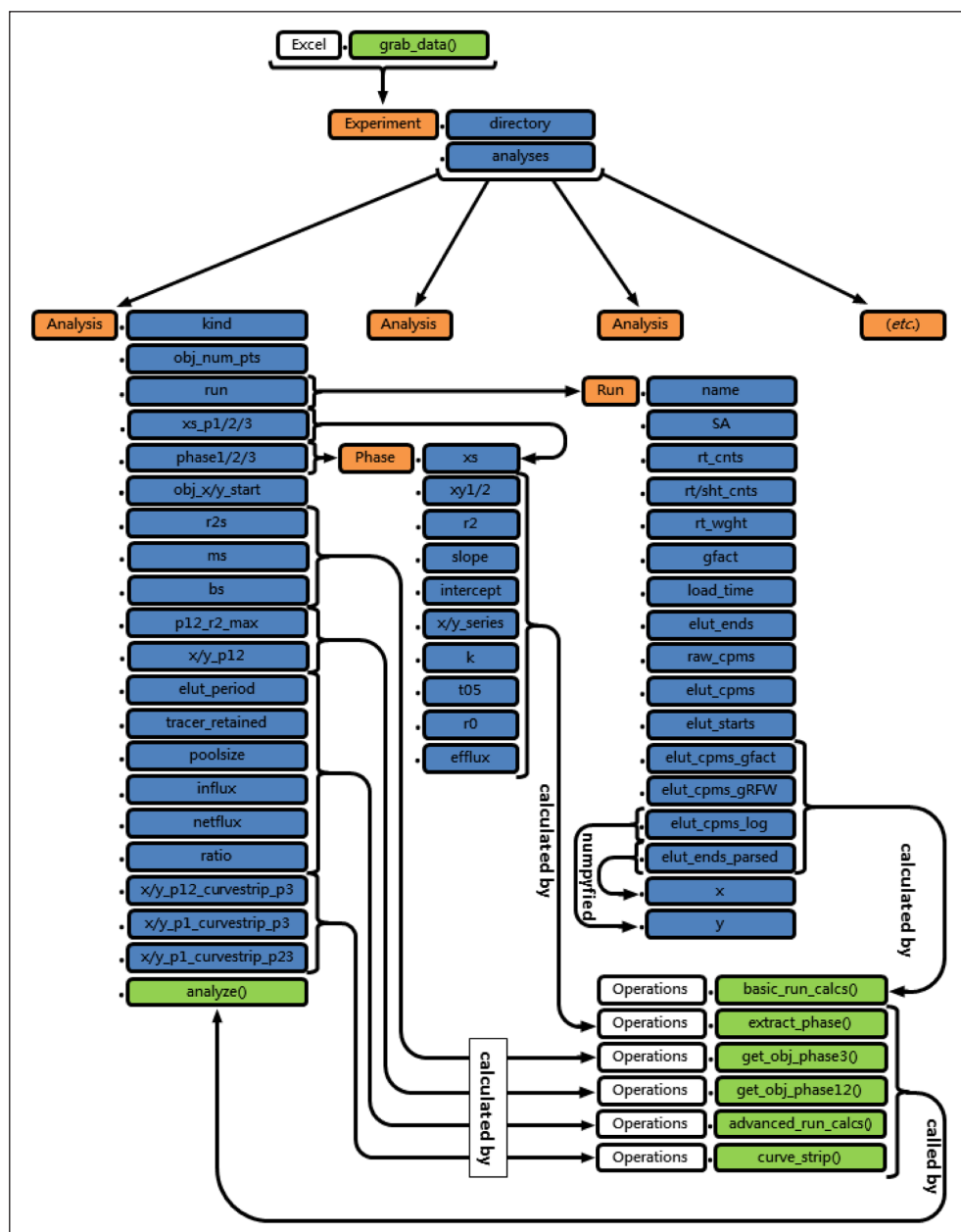


Figure 3: General architecture of vaCATE's back-end. Modules are represented by white boxes, objects by orange boxes, object attributes by blue boxes, and methods/functions by green boxes. Forward slashes ('/') in attribute names are used to indicate the different forms that exist (e.g., `obj_x/y_start` means that both `obj_x_start` and `obj_y_start` attributes exist). Custom objects created by vaCATE can be assumed to belong to the Objects module. Note that this figure is meant to represent a general overview, and not necessarily the sequence in which data is processed.

previewed and its run number relative to the larger data set are displayed in the title bar. Below that, three labelled graphs display a preview of the data extracted from the three phases of the compartmental analysis. In the navigation bar below that, the first four buttons from the left allow users to manipulate how the data is displayed in the preview window, and the following 'left arrow' and 'right arrow' buttons allow users to change which CATE replicate is displayed. The final 'floppy disk' button allows users to output the current analysis of the dataset to an Excel (.xlsx) file.

The areas labelled 'Objective Regression' and 'Subjective Regression' allow users to define and implement their own objective and subjective regressions, respectively (see Types of Regression Models section below). User-defined regressions can be implemented by filling in the appropriate fields in either area and pressing the corresponding 'Draw Objective/Subjective Regression' button. Below this is the 'Propagate Regression' button, which helpfully allows users to apply currently-input regression settings to every CATE replicate being analysed. Finally, the area labelled 'Regression Parameters' contains the data extracted from compartmental analysis of phases that have been, or can be, defined.

Types of Regression Models

Compartmental analysis of data with vaCATE can be done using either objective or subjective regression models. The objective regression model is conducted as described in Siddiqi et al. (1991) [20] and Kronzucker et al. (1999) [36]. Briefly, a linear regression using the last n eluates from the CATE replicate is calculated (where n is the number of points specified in the 'Number of points to use' field;

Figure 4). This is repeated with the last $n + 1$, $n + 2$, etc. eluates and successive coefficients of correlation (R^2) of the regressions are compared. Once the coefficients of correlation decrease three times in a row, the eluate occurring before these three decreases is determined to be the beginning of the phase boundary. Conversely, the subjective regression model allows the user to directly define these boundaries. If the user were to subjectively set the boundaries to be the same as those defined by the objective regression model, then the output parameters of the resulting compartmental analyses would be the same. Boundaries are expressed using the elution times of the eluates (column under cell 'B8' in **Figure 2**; x-axis of any plot in **Figure 4**).

Users wishing to extend the functionality of vaCATE may implement their own algorithm to determine phase boundaries. All that is required of such an algorithm is the generation of tuples which would be used to set the boundaries of each phase ('xs_p1/2/3' attribute; **Figure 3**), as is the case with the subjective and objective regression models discussed above.

Note that the default analysis conducted by vaCATE if the 'Automatically Analyze' option is left selected (**Figure 1**) is an objective regression using the last 8 eluates from each CATE replicate. However, this number of points used to start the objective regression can later be set by the user in the 'Objective Regression' area (see **Figure 4**).

Compartmental Analysis of Phase Parameters

After the 'xs_p1/2/3' attributes of the Analysis object have been set, compartmental analysis can be conducted. This is done by calling the analyse() method on the Analysis object (**Figure 3**), which extracts the compartmental

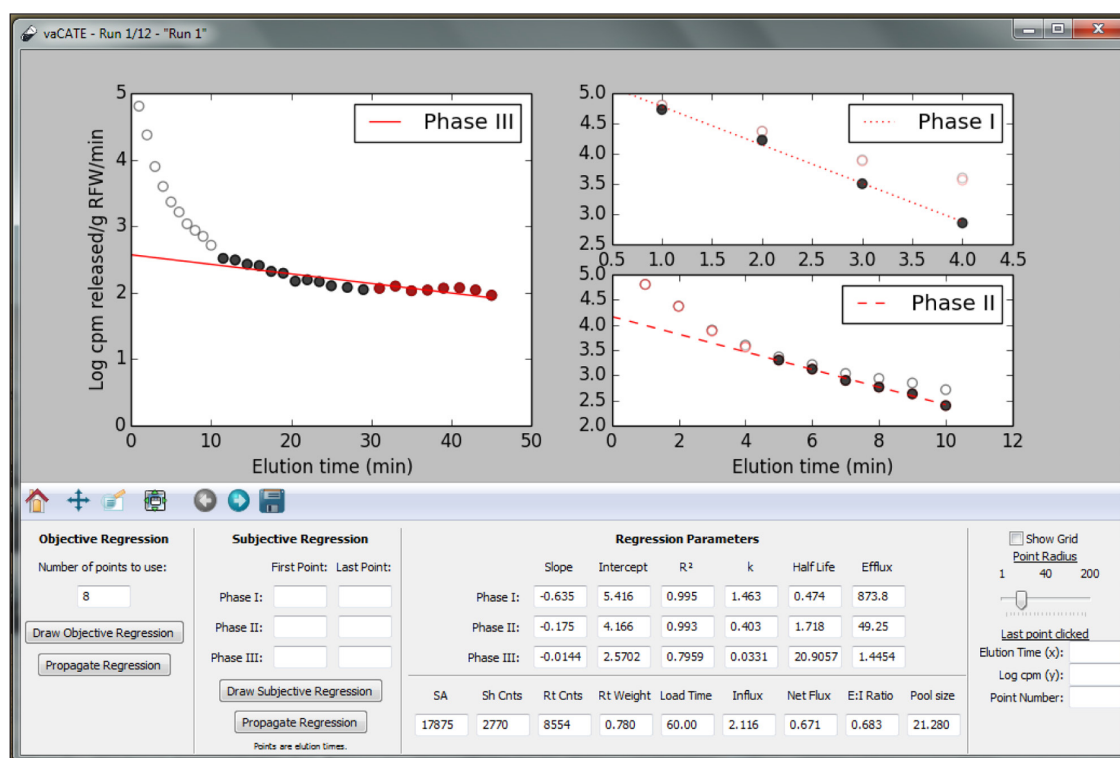


Figure 4: Window displaying the current analysis applied to the data set. This allows users to dynamically change and preview analyses before export to an Excel (.xlsx) file.

analysis of multiple phases in sequential order (later, slower phases before earlier, faster phases). Nesting of “if” statements restricts compartmental analysis to only valid situations (i.e., to phases that are either the slowest, or for which compartmental analysis of later, slower phases has been done), and is recommended if users are using the vaCATE software backend to independently to conduct their own compartmental analysis.

Compartmental analysis of a phase is specifically done by calling the `extract_phase()` function in the Operations module. The Phase objects returned from this are stored as attributes of the Analysis object (Analysis.phase3/2/1; see **Figure 3**). Phases for which compartmental analysis is not valid or not done (see above) are stored as blank phases (with all attributes set to empty strings).

Within the `extract_phase()` method, phase boundaries are converted to indices specific to the data series of the phase that is being analysed (an important error-checking mechanism; see Curve-stripping section below). These indices are used to demarcate elution times and logarithmically-expressed radioisotope release rates specific to the phase in question from the larger x- and y-series (respectively) that were passed to the method. This linear regression is used in conjunction with other data from the CATE replicate to calculate various parameters of the phase required for compartmental analysis (**Table 1**) [1, 17, 20, 37, 38].

Curve-stripping

Slowly-exchanging phases occur simultaneously alongside faster phases. As such, their contribution to total tracer efflux must be removed in order to isolate the faster phases. This is done using a curve-stripping method which is coded into the `curvestrip()` function of the Operations module (**Figures 3 and 5**; lines 298–335). This function is passed four parameters: the x- and y-series of the data that are to be curve-stripped and the slope and intercept of the regression line calculated from the immediately following slower phase. Using the slope and intercept, this slower phase is extrapolated into the range of the faster phase(s) that are to be curve-stripped (**Figure 5**; lines 315–319). The curve-stripping then occurs; that is, the antilog of this extrapolated data from the slower phase is subtracted

from the antilog of the data from the faster phase(s) (**Figure 5**; lines 326–334). The result of this operation is then re-expressed logarithmically.

Here, an important data-validation step is implemented (**Figure 5**; lines 330–332); subtractions that would result in negative logarithmic operations are not added to the curve-stripped x- and y-series that are to be returned. These negative-log operations occur when the extrapolated tracer efflux from the slower phase is ostensibly larger than the cumulative tracer efflux from both this phase and earlier, more rapidly-exchanging phases. While obviously impossible, such data points occasionally come about due to experimental error (CATE is an extremely labour-intensive procedure which can be prone to errors) or to resolution limitations of the detection equipment. If the problem is limited to a small number of data points (at most two), the CATE replicate in question is usually minorly affected, and remains representative of the tested conditions. As long as such problems do not consistently afflict the majority of the CATE replicates conducted, affected replicates should be included in experimental results in order to avoid wasting resources and labour.

Accounting for Shifting Data Series'

The data-validation step presented above poses some interesting challenges from a software engineering perspective. Most poignantly, data points used as boundaries for earlier, more rapidly exchanging phases may be removed. Additionally, a sufficient number of data points may be removed such that compartmental analysis is rendered impossible (i.e., the boundaries delineating a phase contain less than two data points). A discussion regarding how these challenges were dealt with is outside the scope of the current work. However, further information can be found in the 'README.md' file found on vaCATE's github page.

Data Output

After regressions on the examined data set have been satisfactorily configured by the user, the current set of analyses can be exported into a spread sheet by clicking the 'floppy disk' button in the navigation bar (see Dynamic Preview of Data section above). The outputted

Table 1: Names, descriptions and calculations of relevant parameters involved in compartmental analysis by tracer efflux (CATE). For more in-depth discussion regarding these parameters and their calculations, readers are directed to relevant works in the literature [1, 17, 20, 37, 38].

Parameter name in			Description	Calculation
Literature	GUI*†	Backend†		
S_o	SA	SA	Specific activity of loading solution	Radioactivity of loading solution (cpm/mL)
k	k	k	Rate constant	slope of regression line $\times 2.303$
$t_{0.5}$	Half-Life	t05	Half-life of exchange	$0.693/k$
R_o	—	r0	Rate of radioisotope release at t_o	Antilog of x-intercept
ϕ_{co}	Efflux	efflux	Efflux from compartment	R_o/S_o
ϕ_{net}	Net Flux	netflux	Net flux	(Root + shoot radioactivity)/ S_o /root weight
ϕ_{oc}	Influx	influx	Influx	$\phi_{co} \times \phi_{net}$

* Graphical user interface; † of vaCATE.

```

298 def curvestrip(x_series, y_series, slope, intercept):
299     """Create a series of data that has been curve-stripped...."""
315     # Calculating later phase data extrapolated into earlier phase
316     extrapolated_raw = []
317     for item in x_series:
318         extrapolated_x = (slope*item) + intercept
319         extrapolated_raw.append(extrapolated_x)
320
321     # Antilog extrapolated p3 data and p1/2 data, subtract them, and re-log them
322     # Containers for curve-stripped p1/2 data
323     y_curvestrip = []
324     x_curvestrip = [] # need x series because x_curvestrip maybe != x_series
325
326     for index, item in enumerate(y_series):
327         antilog_orig = 10 ** item
328         antilog_reg = 10 ** extrapolated_raw[index]
329         curvestrip_x_raw = antilog_orig - antilog_reg
330         if curvestrip_x_raw > 0: # We can perform a log operation
331             y_curvestrip.append(math.log10(curvestrip_x_raw))
332             x_curvestrip.append(x_series[index])
333         else: # No log operation possible. Data omitted from series
334             pass
335     return x_curvestrip, y_curvestrip
336

```

Figure 5: The `curvestrip()` method in the Operations module is used to remove the effects of later, slower phases from earlier, faster phases.

spread sheet is saved under the name 'vaCATE Output – (YYYY_MM_DD).xlsx' in the same folder that the data was read from.

Quality control

To ensure vaCATE operates as expected, results from four CATE experiments in rice (*Oryza sativa*), encompassing over 90 replicates, were calculated by hand twice, using phase boundaries as determined by both objective and subjective regression.

The results from manual calculations were used to test every possible parameter determined by vaCATE. Each CATE replicate was subjected to over 60 individual tests (using the unittests module; see Tests.py).

Furthermore, 24 'edge cases' were manually constructed to specifically test situations that were likely to create errors in vaCATE. These situations included, but were not limited to: phases missing data points that corresponded to phase boundaries, phases missing sufficient data points as to render CATE impossible, efflux traces in which objective regression is impossible (i.e., in which the coefficient of correlation never decreases three times in a row; see Types of Regressions section), etc. These 'edge cases' were subject to the same testing process that is described above, and are named according to the situation they are testing.

All of the test cases can be found in the 'Tests' folder where vaCATE was installed or cloned, and are already formatted according to the CATE template file requirements. To see if the software is working, a user should simply select a test file for analysis with vaCATE, after which they should see a preview dialog (similar to **Figure 4**). This, in turn, should create an output Excel file and close upon selection of the 'floppy disc' button. If the user selects a file titled 'Test_Multirun1.xlsx' from one of these folders to be analysed by vaCATE then the resultant output Excel file can be directly compared to an provided example output file found in the same folder that the data was read from.

(2) Availability

Operating system

vaCATE has been tested on Windows 7, 8.1, and 10.

Programming language

Python 2.7.5. Compatibility with Python 3 has not been tested.

Additional system requirements

Any system capable of running the operating systems on which vaCATE has been tested (i.e., Windows 7/8.1/10) should be capable of running vaCATE.

Dependencies

When running vaCATE from the source code, the following dependencies are required for Windows 7:

- Python version 2.7.5.
- matplotlib (version 1.3.1/1.5.3). Used for creating figures displayed by GUI.
- numpy (version 1.8.2). Used for mathematical calculations.
- wx (version 2.8.12.1/3.03). Used to create GUI.
- xlrd (version 0.9.2). Used to read Excel files.
- xlswriter (version 0.6.5). Used to create Excel files.

When testing vaCATE, the following dependencies are required:

- nose (version 1.3.7).
- nose-parameterized (version 0.5.0).

When cloning the vaCATE repository, the above dependencies can be installed using the included 'requirement.txt' file by running the 'pip install -r requirements.txt' command inside command prompt. We recommend that this command be run inside a virtual environment, in line with current industry best practises. When vaCATE is installed and run using the executable setup file, the above dependencies are automatically installed.

List of contributors

The integration of matplotlib and wx was initially implemented using a template created by Eli Bendersky [39].

Software location

Archive

(e.g., institutional repository, general repository) (required – please see instructions on journal website for depositing archive copy of software in a suitable repository)

Name: figshare

Persistent identifier: <https://doi.org/10.6084/m9.figshare.4688503.v2>

Publisher: Rubens Flam-Shepherd

Version published: 1.0

Date published: 10/05/17

Code repository

Name: Github

Identifier: <https://github.com/rubenflamshepherd/vaCATE>

Licence: GNU General Purpose License version 3.0

Date published: 10/05/17

Language

English

(3) Reuse potential

In the presented work, vaCATE is used to automate parameter extraction from up to three phases found in compartmental analysis by tracer efflux (CATE), the task

for which it was specifically written. However, vaCATE is not limited to situations where data is collected using radioisotopes. Parameters can be extracted using data from any compartmental analysis, as long as the measured substance moving into/out of compartments can be detected with sufficient temporal resolution, and the system being modelled involves up to three sequential compartments.

In the field of pharmacokinetics four or more sequential phases can occasionally be seen when conducting compartmental analysis. In this situation, users can easily utilize vaCATE's backend to conduct/automate the analysis they require. To do this, users will need to first define phase boundaries. This can be done using the objective regression provided by vaCATE, or via a custom algorithm defined by the user (see Types of Regression Models section). Once phase boundaries have been established, parameters from the phases can be extracted by calling the `extract_phase()` function. For sequential phases, parameters from the last, most slowly-exchanging phase should be extracted first, and the `curvestrip()` function should be called on the data representing the remaining phases. After this, the process should be repeated with the next slowest phase, and so on. Both the `extract_phase()` and `curve_strip()` functions can be found in the Operations module.

Competing Interests

The authors have no competing interests to declare.

Author Information

1. Rubens Flam-Shepherd
 - Software developer and wrote the manuscript
 - Department of Cell & Systems Biology, University of Toronto, CAN
 - Ruben dot flam dot shepherd at mail dot utoronto dot ca
2. Dev T. Britto
 - Helped with editing the manuscript and provided technical advice
 - Department of Cell & Systems Biology, University of Toronto, CAN
 - Britto at utsc dot utoronto dot ca
3. Herbert J. Kronzucker
 - Helped with editing the manuscript
 - Faculty of Veterinary and Agricultural Sciences, University of Melbourne, AUS
 - Herbert dot kronzucker at unimelb dot edu dot au

References

1. **Jacquez, J A** 1996 *Compartmental analysis in biology and medicine*. 3rd edn. Ann Arbor, Michigan: BioMedware.
2. **Shotwell, M S, Zhou, M and Fissell, W H** 2016 'Optimal design of perturbations for individual two-compartment pharmacokinetic analysis'. *Journal of Biopharmaceutical Statistics*, 26(5): 803–815. DOI: <https://doi.org/10.1080/10543406.2015.1074918>
3. **Eksborg, S** 2013 'Individual and population pharmacokinetic compartment analysis: A graphic procedure for quantification of predictive

- performance'. *Journal of Drug Assessment*, 2(1): 135–140. DOI: <https://doi.org/10.3109/21556660.2013.838569>
4. **Olsen, M J, Young, A P and Ashford, S A** 2012 'TopCAT-Topographical Compartment Analysis Tool to analyze seaciff and beach change in GIS'. *Computers and Geosciences*, 45: 284–292. DOI: <https://doi.org/10.1016/j.cageo.2011.11.007>
 5. **Bjerregaard, P, Bjørn, L, Nørum, U and Pedersen, K L** 2005 'Cadmium in the shore crab *Carcinus maenas*: Seasonal variation in cadmium content and uptake and elimination of cadmium after administration via food'. *Aquatic Toxicology*, 72: 5–15. DOI: <https://doi.org/10.1016/j.aquatox.2004.11.018>
 6. **Kang, Y and Castillo-Chavez, C** 2012 'Multiscale analysis of compartment models with dispersal'. *Journal of Biological Dynamics*, 6(sup2): 50–79. DOI: <https://doi.org/10.1080/17513758.2012.713125>
 7. **Hirabayashim, Y, Yoon, B-I, Tsuboi, I, Huo, Y, Kodama, Y, Kanno, J, Ott, T, Trosko, J E and Inoue, T** 2007 'Membrane Channel Connexin 32 Maintains Lin⁻/c-kit⁺ Hematopoietic Progenitor Cell Compartment: Analysis of the Cell Cycle'. *Journal of Membrane Biology*, 217: 105–113. DOI: <https://doi.org/10.1007/s00232-007-9042-z>
 8. **Røe, K, Aleksandersen, T B, Kristian, A, Nilsen, L B, Seierstad, T, Qu, H, Ree, A H, Olsen, D R and Malinen, E** 2010 'Preclinical dynamic ¹⁸F-FDG PET – tumor characterization and radiotherapy response assessment by kinetic compartment analysis'. *Acta Oncologica*, 49(7): 914–21. DOI: <https://doi.org/10.3109/0284186X.2010.498831>
 9. **Schulze, L M, Britto, D T, Li, M and Kronzucker, H J** 2012 'A pharmacological analysis of high-affinity sodium transport in barley (*Hordeum vulgare* L.): A ²⁴Na⁺/⁴²K⁺ study'. *Journal of Experimental Botany*, 63(7): 2479–2489. DOI: <https://doi.org/10.1093/jxb/err419>
 10. **Hamam, A M, Britto, D T, Flam-Shepherd, R and Kronzucker, H J** 2016 'Measurement of Differential Na⁺ Efflux from Apical and Bulk Root Zones of Intact Barley and *Arabidopsis* Plants'. *Frontiers in Plant Science*, 7(272): 1–8. DOI: <https://doi.org/10.3389/fpls.2016.00272>
 11. **Balkos, K D, Britto, D T and Kronzucker, H J** 2010 'Optimization of ammonium acquisition and metabolism by potassium in rice (*Oryza sativa* L. cv. IR-72)'. *Plant, Cell & Environment*, 33: 23–34. DOI: <https://doi.org/10.1111/j.1365-3040.2009.02046.x>
 12. **Coskun, D, Britto, D T, Kochian, L V and Kronzucker, H J** 2016 'How high do ion fluxes go? A re-evaluation of the two-mechanism model of K⁺ transport in plant roots'. *Plant Science*, 243: 96–104. DOI: <https://doi.org/10.1016/j.plantsci.2015.12.003>
 13. **Wang, L and Quintiere, J G** 2009 'An analysis of compartment fire doorway flows'. *Fire Safety Journal*, 44(5): 718–731. DOI: <https://doi.org/10.1016/j.firesaf.2009.02.001>
 14. **Britto, D T and Kronzucker, H J** 2003 'Trans-stimulation of ¹³NH₄⁺ efflux provides evidence for the cytosolic origin of tracer in the compartmental analysis of barley roots'. *Functional Plant Biology*, 30(12): 1233–1238. DOI: <https://doi.org/10.1071/FP03147>
 15. **Britto, D T, Szczerba, M W and Kronzucker, H J** 2006 'A new, non-perturbing, sampling procedure in tracer exchange measurements'. *Journal of Experimental Botany*, 57(6): 1309–1314. DOI: <https://doi.org/10.1093/jxb/erj105>
 16. **Britto, D T and Kronzucker, H J** 2012 'Isotope techniques to study kinetics of Na⁺ and K⁺ transport under salinity conditions'. In: Shabala, S and Cuin, T A (eds.), *Methods in Molecular Biology*, 389–398. Humana Press. DOI: <https://doi.org/10.1007/978-1-61779-986-0>
 17. **Walker, N A and Pitman, M G** 1976 'Measurement of Fluxes across Membranes'. In: Lüttge, U and Pitman, M G (eds.), *Encyclopedia of Plant Physiology*, Vol. 2 Part a: 93–126. Berlin, Germany: Springer-Verlag. DOI: https://doi.org/10.1007/978-3-642-66227-0_5
 18. **Lee, R B and Clarkson, D T** 1986 'Nitrogen-13 studies of nitrate fluxes in barley roots I. Compartmental analysis from measurements of ¹³N efflux'. *Journal of Experimental Botany*, 37(185): 1753–1767. DOI: <https://doi.org/10.1093/jxb/37.12.1753>
 19. **Britto, D T, Ebrahimi-Ardebili, S, Hamam, A M, Coskun, D and Kronzucker, H J** 2010 '⁴²K analysis of sodium-induced potassium efflux in barley: Mechanism and relevance to salt tolerance'. *New Phytologist*, 186: 373–384. DOI: <https://doi.org/10.1111/j.1469-8137.2009.03169.x>
 20. **Siddiqi, M Y, Glass, A D M and Ruth, T J** 1991 'Studies of the uptake of nitrate in barley III. Compartmentation of NO₃⁻'. *Journal of Experimental Botany*, 42(244): 1455–1463. DOI: <https://doi.org/10.1104/pp.99.4.1582>
 21. **Kronzucker, H J, Szczerba, M W and Britto, D T** 2003 'Cytosolic potassium homeostasis revisited: ⁴²K-tracer analysis in *Hordeum vulgare* L. reveals set-point variations in [K⁺]. *Planta*, 217: 540–546. DOI: <https://doi.org/10.1007/s00425-003-1032-5>
 22. **Britto, D T and Kronzucker, H J** 2001 'Can unidirectional influx be measured in higher plants? A mathematical approach using parameters from efflux analysis'. *New Phytologist*, 150: 37–47. DOI: <https://doi.org/10.1046/j.1469-8137.2001.00080.x>
 23. **Kronzucker, H J, Siddiqi, M Y and Glass, A** 1995 'Analysis of ¹³NH₄⁺ Efflux in Spruce Roots: A Test Case for Phase Identification in Compartmental Analysis'. *Plant Physiology*, 109: 481–490. DOI: <https://doi.org/10.1104/pp.109.2.481>
 24. **Coskun, D, Britto, D T and Kronzucker, H J** 2010 'Regulation and mechanism of potassium release from barley roots: an *in planta* ⁴²K⁺ analysis'. *New Phytologist*, 188: 1028–1038. DOI: <https://doi.org/10.1111/j.1469-8137.2010.03436.x>
 25. **Coskun, D, Britto, D T, Li, M, Becker, A and Kronzucker, H J** 2013 'Rapid Ammonia Gas Transport Accounts for Futile Transmembrane Cycling under NH₃/NH₄⁺ Toxicity in Plant Roots'. *Plant Physiology*,

- 163: 1859–1867. DOI: <https://doi.org/10.1104/pp.113.225961>
26. **Coskun, D, Britto, D T, Li, M, Oh, S and Kronzucker, H J** 2013 'Capacity and plasticity of potassium channels and high-affinity transporters in roots of barley and *Arabidopsis*'. *Plant Physiology*, 162: 496–511. DOI: <https://doi.org/10.1104/pp.113.215913>
 27. **Behl, R and Jeschke, W D** 1982 'Potassium Fluxes in Excised Barley Roots'. *Journal of Experimental Botany*, 33(135): 584–600. DOI: <https://doi.org/10.1093/jxb/33.4.584>
 28. **Jeschke, W D** 1982 'Shoot-Dependent Regulation of Sodium and Potassium Fluxes in Roots of Whole Barley Seedlings'. *Journal of Experimental Botany*, 33(135): 601–618. DOI: <https://doi.org/10.1093/jxb/33.4.601>
 29. **Memon, A R, Saccomani, M and Glass, A D M** 1985 'Efficiency of Potassium Utilization by Barley Varieties: The Role of Subcellular Compartmentation'. *Journal of Experimental Botany*, 36(173): 1860–1876. DOI: <https://doi.org/10.1093/jxb/36.12.1860>
 30. **Wang, M Y, Siddiqi, M Y, Ruth, T J and Glass, A D M** 1993 'Ammonium Uptake by Rice Roots I. Fluxes and Subcellular Distribution of $^{13}\text{NH}_4^+$ '. *Plant Physiology*, 103: 1249–1258. DOI: <https://doi.org/10.1104/pp.103.4.1249>
 31. **Coskun, D, Britto, D T, Hamam, A M and Kronzucker, H J** 2014 'Measuring fluxes of mineral nutrients and toxicants in plants with radioactive tracers'. *Journal of Visualized Experiments*, 90: e51877. DOI: <https://doi.org/10.3791/51877>
 32. **Barrett, P H R, Bell, B M, Cobelli, C, Golde, H, Schumitzky, A, Vicini, P and Foster, D M** 1998 'SAAM II: Simulation, Analysis, and Modeling Software for tracer and pharmacokinetic studies'. *Metabolism*, 47(4): 484–492. DOI: [https://doi.org/10.1016/S0026-0495\(98\)90064-6](https://doi.org/10.1016/S0026-0495(98)90064-6)
 33. **Zhang, Y, Huo, M, Zhou, J and Xie, S** 2010 'PKSolver: An add-in program for pharmacokinetic and pharmacodynamic data analysis in Microsoft Excel'. *Computer Methods and Programs in Biomedicine*, 99(3): 306–314. Elsevier. DOI: <https://doi.org/10.1016/j.cmpb.2010.01.007>
 34. **Tanswell, P and Koup, J** 1993 'TopFit: A PC-based pharmacokinetic/pharmacodynamic data analysis program'. *International Journal of Clinical Pharmacology, Therapy, and Toxicology*, 31(10): 514–520.
 35. **Britto, D T and Kronzucker, H J** 2013 'Flux Measurements of Cations Using Radioactive Tracers'. In: Walker, J M (ed.), *Methods in Molecular Biology*, 161–170. Humana Press. DOI: https://doi.org/10.1007/978-1-62703-152-3_10
 36. **Kronzucker, H J, Siddiqi, M Y, Glass, A D M and Kirk, G J D** 1999 'Nitrate-Ammonium Synergism in Rice. A Subcellular Flux Analysis'. *Plant physiology*, 119(3): 1041–1046. DOI: <https://doi.org/10.1104/pp.119.3.1041>
 37. **Malagoli, P, Britto, D T, Schulze, L M and Kronzucker, H J** 2008 'Futile Na^+ cycling at the root plasma membrane in rice (*Oryza sativa* L.): Kinetics, energetics, and relationship to salinity tolerance'. *Journal of Experimental Botany*, 59(15): 4109–4117. DOI: <https://doi.org/10.1093/jxb/ern249>
 38. **Coskun, D, Britto, D T and Kronzucker, H J** 2014 'The physiology of channel-mediated K^+ acquisition in roots of higher plants'. *Physiologia Plantarum*, 151(3): 305–312. DOI: <https://doi.org/10.1111/ppl.12174>
 39. **Bendersky, E** 2008 *matplotlib with wxPython GUIs*. Available at: <http://eli.thegreenplace.net/2008/08/01/matplotlib-with-wxpython-guis/>.

How to cite this article: Flam-Shepherd, R, Britto, D T and Kronzucker, H J 2018 vaCATE: A Platform for Automating Data Output from Compartmental Analysis by Tracer Efflux. *Journal of Open Research Software*, 6: 21. DOI: <https://doi.org/10.5334/jors.175>

Submitted: 24 April 2017

Accepted: 08 August 2018

Published: 17 August 2018

Copyright: © 2018 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.