Published online 12 May 2016 *Nucleic Acids Research, 2016, Vol. 44, Web Server issue* ***W205–W211***
doi: 10.1093/nar/gkw415

PASMet: a web-based platform for prediction, modelling and analyses of metabolic systems

Kansuporn Sriyudthsak[1], Ramon Francisco Mejia[1], Masanori Arita[1,2] and Masami Yokota Hirai[1,*]

[1]RIKEN Center for Sustainable Resource Science, Yokohama, Kanagawa 230-0045, Japan and [2]National Institute of Genetics, Mishima, Shizuoka 411-8540, Japan

Received February 12, 2016; Revised April 22, 2016; Accepted May 01, 2016

ABSTRACT

PASMet (Prediction, Analysis and Simulation of Metabolic networks) is a web-based platform for proposing and verifying mathematical models to understand the dynamics of metabolism. The advantages of PASMet include user-friendliness and accessibility, which enable biologists and biochemists to easily perform mathematical modelling. PASMet offers a series of user-functions to handle the time-series data of metabolite concentrations. The functions are organised into four steps: (i) Prediction of a probable metabolic pathway and its regulation; (ii) Construction of mathematical models; (iii) Simulation of metabolic behaviours; and (iv) Analysis of metabolic system characteristics. Each function contains various statistical and mathematical methods that can be used independently. Users who may not have enough knowledge of computing or programming can easily and quickly analyse their local data without software downloads, updates or installations. Users only need to upload their files in comma-separated values (CSV) format or enter their model equations directly into the website. Once the time-series data or mathematical equations are uploaded, PASMet automatically performs computation on server-side. Then, users can interactively view their results and directly download them to their local computers. PASMet is freely available with no login requirement at http://pasmet.riken.jp/ from major web browsers on Windows, Mac and Linux operating systems.

INTRODUCTION

Computational approaches have become indispensable for synthetic biology and metabolic engineering research. In general, experimental data are subjected to statistical analyses and mathematical modelling to comprehensively understand the related biological and biochemical systems. For time-series data, the process includes prediction of metabolic pathways and their regulation using the estimated causality from quantitative data, simulation of probable metabolic behaviours, and analysis of system characteristics using mathematical models. To this end, various theories and algorithms have been proposed to process biological data systematically. A number of visualisation software packages for statistical analyses have also been developed for both academic and commercial purposes. Although these resources allow researchers to analyse their data themselves, computational tasks such as mathematical modelling are not easily accessible for researchers who may not have a strong mathematical background and computer programming skills.

Currently, there are several well-known programming languages for statistical analysis and mathematical modelling. One is the R Project (1), which provides numerous packages to build statistical analysis programs with strong support for visualisation. Other products include Matlab (2) and Python (3), which offer high-performance scientific computing to develop software programs for mathematical modelling. In these languages, however, basic knowledge of software engineering is required to write accurate and efficient codes. Custom software programs to simulate and analyse biochemical networks (4–8) also require computer skills for installation and setup as well as expertise and additional effort to exploit their functionalities. Web-based applications such as JWS Online (9) and WebCell (10) lowered the hurdle to work on kinetic modelling, dynamic simulation and system analysis in details. For detailed analysis of metabolic networks, SoftCADs (11) now offers dynamic sensitivity analysis for predicting the dynamic responses of metabolite concentrations.

With an intention to offer a series of functions to easily handle the time-series data of metabolite concentrations, we present PASMet (Prediction, Analysis and Simulation of Metabolic networks), a user-friendly web-based platform to both predict pathways and construct mathematical models to systematically analyse metabolic systems. PASMet

*To whom correspondence should be addressed. Tel: +81 45 503 9491; Fax: +81 45 503 9489; Email: masami.hirai@riken.jp

provides an interface to access various algorithms for predicting probable regulatory mechanisms along with generating mathematical models from time-series data, simulating metabolic behaviours and determining metabolic bottlenecks in the model.

IMPLEMENTATION

PASMet was written in Python with server-side algorithms using NumPy (12) and SciPy (13) for modelling, StatsModels (14) for statistical analysis and Matplotlib (15) for graphical plots. The client-side interface was developed as a web-based application built on top of the Django framework (16). Users without enough knowledge of computing or programming can easily and quickly analyse their local data without software downloads, updates or installations. Users are only required to upload their file in comma-separated values (CSV) format or type their model equations directly into the website. Then, PASMet automatically performs computations in a server-side environment. Accordingly, users can interactively view the results and directly download them to their local computers. For stability checking, load tests were performed using Locust (17) by simulating 50 users performing five requests per second. To support long-running tasks, an asynchronous task queue was also implemented on the front-end using Ajax, while the back-end services used Celery (18) and RabbitMQ (19).

PROGRAM DESCRIPTION AND METHODS

PASMet provides four functionalities (Figure 1): (i) Prediction, (ii) Construction, (iii) Simulation and (iv) Analysis. The Prediction function is divided into two principal approaches (Prediction 1 and 2). All functions can be performed independently or in combination. Documentations and tutorial with detailed examples are provided on our website. Sample files of time-series data and three mathematical models, a lactate model, an aspartate model and a simple linear model using Michaelis–Menten equations, are also available for download. To assist new users, these files can be uploaded directly for each function on the website.

To experience and test the prediction of a metabolic pathway and its regulatory mechanism (Prediction 1 and 2 functions), a 48-point time-series data set of the metabolite concentrations for the glycolysis pathway in *Lactococcus lactis* MG1363 (20–22) (lactate model) was prepared. The data for five metabolites in the glycolysis pathway are included: glucose (Glc), glucose 6-phosphate (G6P), fructose bisphosphate (FBP), lactate (Lac) and acetate (Ace). To test the Construction function, a simple but stiff model of the aspartate-family amino acid biosynthesis pathway in *Arabidopsis thaliana* (23) (aspartate model) was prepared. The 21-point time-series data of metabolite concentrations in this biosynthesis pathway encompass seven metabolites: aspartate-4-phosphate (A4P, x_1), aspartate-semialdehyde (ASA, x_2), lysine (LYS, x_3), homoserine (HS, x_4), *O*-phospho-homoserine (OPH, x_5), threonine (THR, x_6) and isoleucine (ILE, x_7). In the Simulation and Analysis functions, the model equations for the lactate, aspartate and simple linear models are available.

Prediction 1

A comprehensive understanding of metabolic networks and their regulatory mechanisms is important in genetic and metabolic engineering research. However, a lack of information could result in not identifying previously unknown pathways, which may lead to an insufficient or incorrect understanding of the network of interest. The *in silico* prediction of a probable pathway has the potential to identify such unknown pathways. One of the computational methods to predict a probable pathway is by combining causal relationships among metabolites, which can be estimated from time-series data of metabolite accumulation. Two common approaches applying statistical methods for generating a causal network are dynamic Bayesian network inference and Granger causality test. Both approaches have their own advantages and theoretical limits and their predictive performances depend on the nature of data and way to process raw data. For example, the dynamic Bayesian network inference is rather suitable for time-series data of short length (small number of data point) whereas the Granger causality test is suitable for time-series data of long length (24).

The 'Prediction 1' function was built to support prediction of a probable pathway using time-series data by Granger causality test (25). Users can predict a metabolic pathway using their time-series data of metabolite concentrations without prior information on pathways and their regulation. The prediction begins by preparing time-series data. In the data file, the first row contains the metabolite names, followed by their concentrations in the subsequent rows. If files are created in a spreadsheet, they should be saved in CSV format. In terms of the calculation in Prediction 1, there is no limit on the number of metabolites, so it is possible to generate a causality network by using data with hundreds of metabolites. However, it is ideal to use less than 20 metabolites for the calculation to appropriately predict a reasonable network. Also, the time-series data of each metabolite concentration should show dynamic change. This is because the algorithm theoretically works well with data which show apparent changes and imply consecutive conversion of a metabolite (or substrate of an enzyme) to subsequent metabolites (or products).

For predicting a probable pathway, Prediction 1 provides an option for processing time-series data before estimating a causal relationship and formulates the predicted relationship into probable pathways. Two data processing methods, which are Average and Principal Component Analysis (PCA), are provided. The selection of the method depends on the nature of the data. PCA is suitable for users who want to observe significant changes throughout the data or when the data have a high standard deviation. In other general cases, processing using Average is recommended. Three data smoothing methods are also available for refining time-series data with missing values or insufficient data points. These methods allows users to smooth data containing high deviations, as well as to generate more data points from original data before performing the Granger causality test. The spline and polynomial smoothing methods are recommended for data that fit a sine-shaped curve. If the number of data points is large (i.e. more than 20),

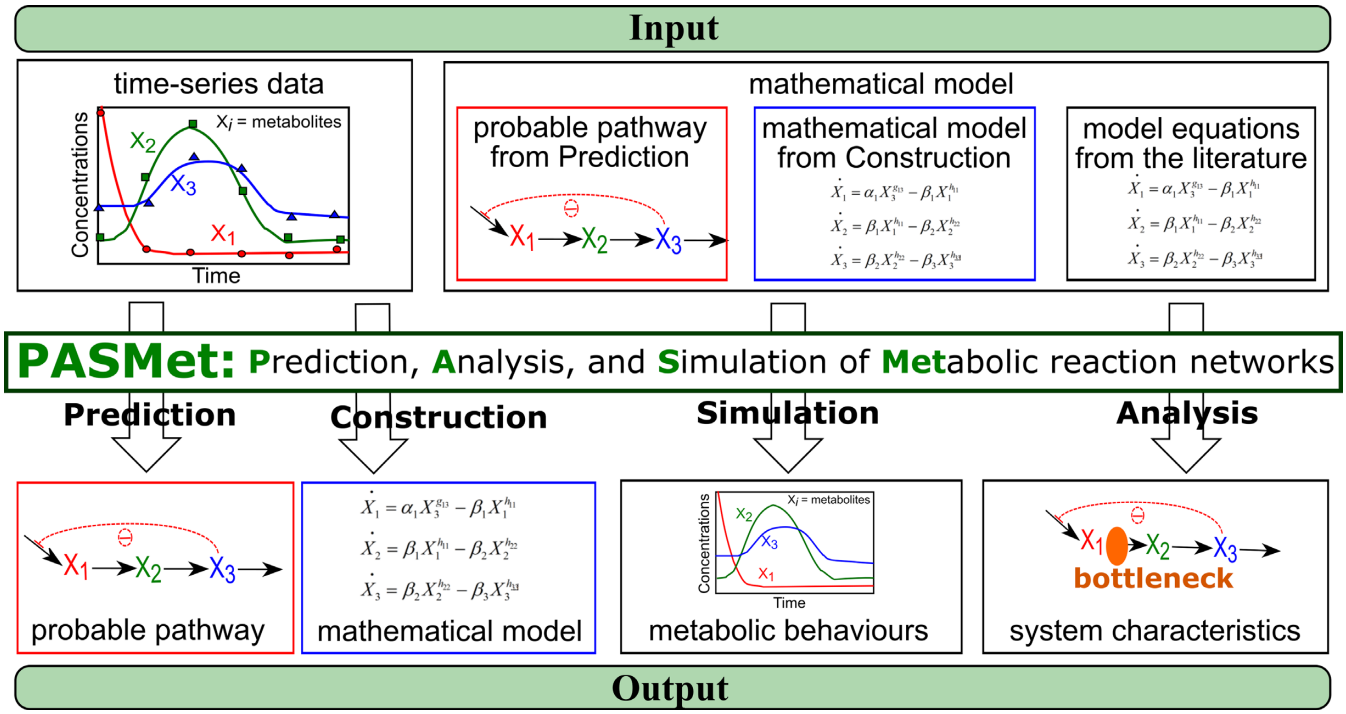

Figure 1. Processing overview of the functions of PASMet, including Prediction, Construction, Simulation and Analysis.

locally estimated scatterplot smoothing (LOESS) is recommended. Once all prediction parameters are specified, Prediction 1 performs Granger causality tests between all pairs of metabolites. Combinations with a statistically significant value are chosen for creating a probable network. The processed data and probable metabolic pathways are visualised using Matplotlib and NetworkX (26), and the calculated results become available for download. The prediction can be repeated many times with different parameters without requiring re-uploads.

As an example, we evaluated the time-series data of metabolite concentrations in the glycolysis pathway of *L. lactis*. Since there was only one experimental replicate in this data, the processing type (PCA or Average) did not affect the result. We also did not perform data smoothing. Given these conditions with *p*-value set to be 0.01, PASMet calculated Granger causality among the five metabolites and predicted that Glc is converted to G6P, then FBP, and finally Ace, as shown in Figure 2A. This result agrees with the glycolysis pathway widely accepted in the literature and databases. Prediction 1 also predicted that Lac affects the concentrations of G6P and FBP. More details on the regulatory mechanisms are provided in Prediction 2.

Prediction 2

Although Prediction 1 suggests the causal relationship among metabolites, it does not prune the regulatory information. The 'Prediction 2' function is designed for specifying regulations, based on the time-series data of metabolite concentrations. To this end, the BST-loglem method (27) is employed to evaluate the relationships by two criteria of causal effects. One is the significant causality on influx for each metabolite (focused metabolite) using the Granger causality test. At least one input is considered essential for each metabolite in the given model. The other is the 'influx/efflux causalities' for each metabolite, estimated from the S-system equations in the framework of biochemical systems theory (BST) (28). The latter 'influx/efflux causalities' are obtained by an automated parameter estimation on the whole model, and weaker causal relationships on influx/efflux of each metabolite can be manually deleted stepwise until a sufficiently simple regulatory mechanism is obtained. There are two rules in the deletion step. First, an interaction that has the strongest significant causality for each metabolite should not be removed from the beginning state. Second, the efflux from the focused metabolite should not be removed even though it may be recommended due to its weakest causation. In case there are a large number of metabolites to be considered, Prediction 2 may be used in combination with Prediction 1.

The preparation of time-series data of metabolite concentrations for Prediction 2 is the same as described for Prediction 1. Although our algorithm can handle an unlimited number of metabolites, we recommend users to supply less than 10 metabolites. Once a time-series data file is uploaded, a focused metabolite is selected. For the calculation of causal effects, the StatsModel and the Levenberg–Marquardt algorithms (29) via the 'optimise' package of SciPy are used for statistical calculations and parameter estimation for mathematical modelling, respectively. After removing a relationship having the weakest causation on fluxes of the focused metabolite, a probable network for each iteration is visualised using NetworkX.

To illustrate, we used the same data set as in Prediction 1, which is the time-series data of metabolites related to

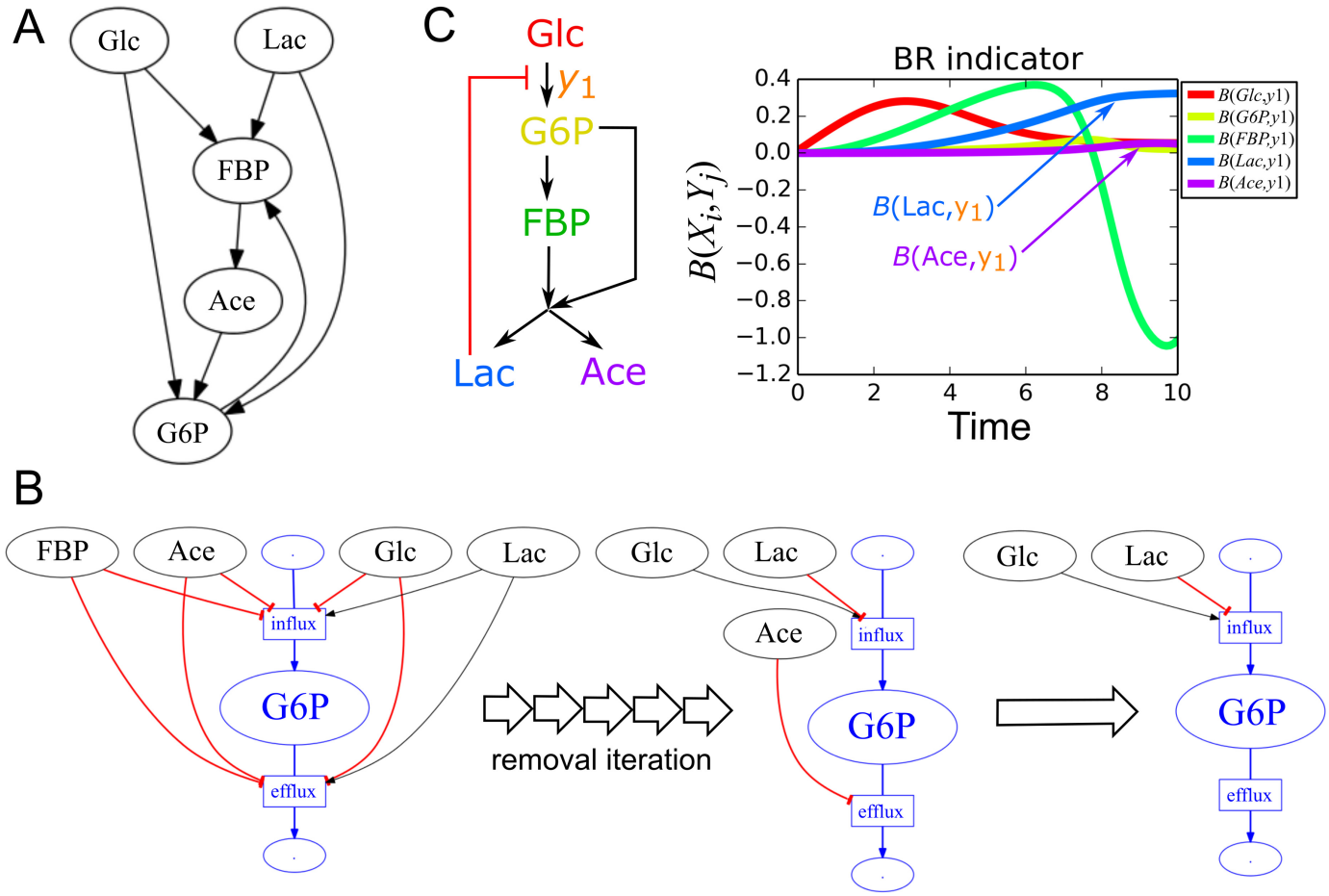

Figure 2. Examples of some of the graphical outputs and analyses available from PASMet. (**A**) Probable metabolic pathway calculated from Prediction 1 using the lactate model. (**B**) Probable metabolic pathway and its regulation related to G6P calculated from Prediction 2. The dot in an ellipse indicates an unspecified metabolite. (**C**) Metabolic pathway for lactate model and bottleneck ranking indicator calculated with the Analysis function.

the glycolysis pathway of *L. lactis*. Since Prediction 1 determined that Lac affects the concentration of G6P and FBP, G6P was selected as a focused metabolite for discovering a probable regulatory mechanism. Figure 2B shows that all considered metabolites are suggested in the first iteration to have effects on flux to/from G6P (the focused metabolite). After removing the weakest causal relationships on flux to/from the focused metabolite for five times, only four possible regulations are left. These include a positive regulation from Glc on influx of G6P, negative regulation from Lac on influx of G6P, negative regulation from Ace on efflux of G6P, and efflux of G6P. Among these four, negative regulation from Ace on efflux of G6P is suggested to have the weakest effect because it has the lowest absolute value. Thus, after this relationship is removed, Prediction 2 predicted that lactate is a probable inhibitory regulator of the formation of G6P (30). The parameter values indicating causal relationships obtained from the Prediction 2 are available for download on the website. The combination of parameter values obtained from the calculations of many focused metabolites related as a network can also be used to construct a mathematical model to aid in the comprehensive understanding of the metabolic reaction network using the Simulation and Analysis steps.

Construction

The 'Construction' function provides a simple algorithm to estimate parameter values for constructing a mathematical model from time-series data of metabolite concentrations. This function uses the Levenberg–Marquardt algorithm coupled with an ordinary differential equation (ODE) solver via the 'optimise' and 'odeint' packages of SciPy (13) to estimate model parameters. In addition to time-series data, Construction also needs information on the metabolic pathway and its regulation to symbolically build mathematical formulations. Although the Construction is applicable to estimate parameters of kinetic equations such as Michaelis–Menten kinetics, its performance is predominant for equations which contain a small number of parameters such as the S-system equations on the basis of BST, especially, simplified equations using the PENDISC method (23). Although there is no input limits on the number of parameters to be estimated, it is recommended to decrease the number of parameters as much as possible.

Users can construct a mathematical model by uploading two CSV files, one for time-series data and the other for model equations. The time-series data should be prepared in the same format as in Prediction 1. Model equations

with initial values for parameter estimation are prepared by setting the number of dependent variables (ND) (i.e. the number of metabolites), the number of independent variables (NI) (i.e. the number of parameters to be estimated), and simulation conditions such as start time, end time and intervals. In addition, the initial values of metabolite concentrations and estimation parameters are necessary. Normally, the initial values of metabolite concentrations are taken from experimental data, whereas the initial values for parameter estimation may be simply set to 10 or 1. Finally, users set formulas for model equations in the form of differential equations. More information on the simple process for setting up equations is provided in our previous publication (23), and examples of various mathematical equations are also available from in the website. Once the Construction is performed, parameter values are estimated and the fitting results are visualised by Matplotlib. Users can use the constructed model which can be downloaded for further simulation of metabolic behaviours or analysis of metabolic systems in the Simulation and Analysis steps.

The aspartate model was chosen as a case study for this function. This model has two parameters to be estimated. Once the data and model files were uploaded, Construction estimated values for the two parameters by fitting calculated lines to the time-series data of seven metabolites simultaneously. The results showed that even for this stiff model, PASMet could provide parameter values with at least five digits of accuracy, which is comparable to those obtained by the in-house parameter estimation algorithms integrating the Levenberg–Marquardt method and the highly accurate differentiation method (11).

Simulation

The 'Simulation' function enables researchers, especially biologists and biochemists, to utilise a mathematical model represented by ODE for observing the dynamic behaviours of metabolite concentrations over time. Simulation also provides a mode for setting up the initial values of metabolite concentration so that users can perform *in silico* metabolic perturbations. To perform simulations, a mathematical model is necessary. Users may use a kinetic model available from the literature, a model constructed by the Construction or a model with simple power-law formulations with most parameters set to unity according to the U-system approach (31). The Simulation contains various methods to solve ODEs (32) of metabolite concentrations. The methods from the SciPy ODE integration package include the Runge–Kutta method, Adams method and backward differential formula (BDF). Users can select a calculation method based on characteristics of their models. Runge–Kutta is a simple method for solving ODEs, whereas Adams and BDF are suitable for solving non-stiff and stiff problems, respectively. The methods also include the LSODA library (33), which automatically switches between non-stiff (Adams) and stiff (BDF) methods depending on system behaviours.

Simulation provides four options. The first and second options are for the direct input of equations for metabolites (ODEs), and that for metabolites and fluxes (enzymatic reactions), respectively. In these two options, a simulation starts by selecting the calculation method and the number of metabolites. Once the number of metabolites is determined, the names and initial values of each metabolite are entered. Users then write down equations as well as edit start time, end time and intervals for examining probable metabolic behaviours. In 'Direct inputs of equations for metabolites', the equations can be either enter whole ODEs or just combined flux equations for each metabolite. In the 'Direct inputs of equations for metabolites and fluxes', equations for enzymatic reactions of each flux need to be filled out separately before balancing them in an equation for each metabolite. In both options, parameter values should be written down explicitly in equations. Once the model equations are determined, a quick simulation is performed and visualised by Matplotlib. Users can also re-simulate dynamic behaviours by changing a condition or the model equations. As examples, the lactate and aspartate models are provided in the 'Direct inputs of equations for metabolites' while the simple linear model using Michaelis–Menten equations is provided in 'Direct inputs of equations for metabolites and fluxes'.

The third and fourth options are for uploading a model file in CSV and SBML (Systems Biology Markup Language) formats, respectively. In these options, a model file can be directly uploaded without any input limits on the number of metabolites. Once a simulation is performed, calculated results and the model file parsing to another format are available for download. A model in CSV-format model is converted to SBML-format model and vice versa. It is to be noted that although PASMet generates an SBML model using the parser in the libSBML Python API (34), it does not provide full SBML support, especially for proper graphical visualisation. This is because some kinds of kinetic models such as a model using S-system equations are approximated and sometimes do not precisely separate flux equations according to mass balance. In case users require more proper SBML models, the 'Direct inputs of equations for metabolites and fluxes' option for building a kinetic model is provided.

Analysis

PASMet also offers the 'Analysis' function for a deeper analysis of a metabolic reaction network using mathematical models. The analysis underlies three indicators: sensitivity, logarithmic gain and the bottleneck ranking (BR) indicator (35), which are defined as $S(X_i, Y_j)$, $L(X_i, Y_j)$ and $B(X_i, Y_j)$, respectively, where X_i and Y_j represent metabolite concentrations and enzyme activity. Sensitivity analysis is typically used for observing robustness and uncertainty in the metabolic behaviours of a mathematical model. Logarithmic gain is the relative sensitivity defined as the change in a dependent variable (metabolite concentration) in response to an infinitesimal change in an independent variable (enzyme activity). The BR indicator is defined as the product of dynamic logarithmic gain and metabolite concentration and is used in the quantitative determination of bottleneck enzymes. For variable parameters such as enzyme activities in the model, the BR indicator can identify the effects of metabolic perturbation by quantitative comparison of metabolite concentrations, which dynamically change in response to parameter changes. Sensitivity results in PASMet

are analytically obtained by a direct differential method (36) in which the sensitivity equations are symbolically obtained by SymPy (37).

In Analysis, users can elucidate dynamic characteristics of a metabolic system by setting a parameter value that refers to enzyme activity. First, users start from writing down metabolite names, initial values of metabolite concentrations, equations for each metabolite and calculation conditions in the similar way in 'Direct input of equations for metabolites' option in Simulation. Then, users select a number of enzymes to be considered as well as write down names and parameter values of those enzymes. The parameter values of considered enzymes which may be explicitly written down in equations should be replaced by the names of those enzymes. Dynamic behaviours of metabolite concentrations are observed in response to a change of considered enzymes.

As two examples, the lactate and aspartate models are provided. In the lactate model, for instance, the effect of influx to Glc on metabolite concentrations was considered (Figure 2C). The parameter value for this flux was defined as y_1 whereas other parameter values in the model were explicitly filled out in equations. The result of BR indicator indicated that $B(Lac,y_1)$, the BR indicator of Lac in response to an increase of y_1, continuously increases faster than $B(Ace,y_1)$. This implied that Lac is formed more easily than Ace, which was consistent with experimental results (20,38).

DISCUSSION

The PASMet web service has several distinctive features compared to other available mathematical modelling software. Many of the well-known modelling tools contain excellent functions and interfaces for mathematical modelling, but they are not as easy to operate as PASMet. Also, most of the high-functioning kinetic modelling software do not include a function for predicting metabolic networks using only experimental data. Since PASMet is designed as a user-friendly tool for biologists and biochemists, it rather focuses on the Prediction and Simulation functions instead of the Construction function. The Prediction function is intended to support biologists and biochemists in quickly analysing their time-series data of metabolite concentrations to discover an unknown pathway, which can then be further studied. The Simulation and Analysis functions are intended to support users who may not be familiar with mathematical models, but wish to explore the dynamics of metabolic systems from a mathematical modelling aspect. For the Construction function, PASMet does not support global optimisations such as genetic algorithms and simulated annealing for the process of parameter estimation. This is because different levels of mathematical expertise are needed to configure the initial values and boundaries for those methods. Thus, the Construction in PASMet is suitable for building a model of a specific pathway containing less than 10 parameters for simple simulations. Construction does not yet support model construction using large-scale data such as metabolome data. More options to support precise calculations and large-scale data applicable for theoretical researchers will be added in the near future.

CONCLUSIONS

PASMet is a user-friendly web-based tool primarily designed to handle time-series data which contain important information in their network structure. It offers four major functions to predict metabolic pathways, construct a mathematical model, simulate metabolic behaviours and analyse a metabolic system. Each function can be applied independently. The comprehensive mathematical approaches for exploiting time-series data to extract biological insights can be performed without strong background on mathematical modelling. PASMet also has a detailed tutorial document for every function available on the homepage.

ACKNOWLEDGEMENT

The authors thank Dr. Tetsuya Sakurai (RIKEN Center for Sustainable Resource Science/Kochi University) for providing access from RIKEN PRIMe to PASMet, and Mr. Yutaka Yamada (RIKEN Center for Sustainable Resource Science) for good discussions on server maintenance and web development. We also thank Prof. Fumihide Shiraishi (Kyushu University) for fruitful discussions on mathematical modelling.

FUNDING

Japan Science and Technology Agency, CREST (to M.Y.H.); JSPS KAKENHI numbers 25660084 and 25113010 (to M.Y.H.); RIKEN Incentive Research Grant (to K.S.); Database Integration Coordination Program of the NBDC (to R.F.M. and M.A.). Funding for open access charge: JSPS KAKENHI number 25113010 (to M.Y.H.).
Conflict of interest statement. None declared.

REFERENCES

1. R Development Core Team. (2010) *R Foundation for Statistical Computing*, Vienna.
2. The MathWorks, I. (2012) *Natick*, Massachusetts.
3. Oliphant,T. (2007) Python for scientific computing. *Comput. Sci. Eng.*, **9**, 10–20.
4. Alves,R., Antunes,F. and Salvador,A. (2006) Tools for kinetic modeling of biochemical networks. *Nat Biotechnol*, **24**, 667–672.
5. Funahashi,A., Matsuoka,Y., Akiya,J., Morohashi,M., Kikuchi,N. and Kitano,H. (2008) CellDesigner 3.5: A versatile modeling tool for biochemical networks. *Proc. IEEE.*, **96**, 1254–1265.
6. Kent,E., Hoops,S. and Mendes,P. (2012) Condor-COPASI: high-throughput computing for biochemical networks. *BMC Syst. Biol.*, **6**, 91.
7. Slepchenko,B.M., Schaff,J.C., Macara,I. and Loew,L.M. (2003) Quantitative cell biology with the Virtual Cell. *Trends Cell Biol.*, **13**, 570–576.
8. Wolfram Research, I. (2015) *SystemModeler*. Version 4.2 edn. Wolfram Research, Inc., Champaign.
9. Olivier,B.G. and Snoep,J.L. (2004) Web-based kinetic modelling using JWS Online. *Bioinformatics*, **20**, 2143–2144.
10. Lee,D.Y., Yun,C., Cho,A., Hou,B.K., Park,S. and Lee,S.Y. (2006) WebCell: a web-based environment for kinetic modeling and dynamic simulation of cellular networks. *Bioinformatics*, **22**, 1150–1151.
11. Shiraishi,F., Furuta,S., Ishimatsu,T. and Akhter,J. (2007) A simple and highly accurate numerical differentiation method for sensitivity analysis of large-scale metabolic reaction systems. *Math. Biosci.*, **208**, 590–606.
12. Walt,S.v.d., Colbert,S.C. and Varoquaux,G. (2011) The NumPy Array: A structure for efficient numerical computation. *Comput. Sci. Eng.*, **13**, 22–30.

13. Jones,E., Oliphant,T., Peterson,P. and others. (2001-) SciPy: Open source scientific tools for python. http://www.scipy.org (11 January 2014, date last accessed).
14. Seabold,J.S. and Perktold,J. (2010) Statsmodels: Econometric and statistical modeling with Python. *Proc. 9th Python Sci. Conf.*, **2010**, 57–61.
15. Hunter,J.D. (2007) Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.*, **9**, 90–95.
16. Django (Version 1.6). (2013) Computer Software. http://www.djangoproject.com.
17. Heyman,J., Bystrom,C., Hamren,J. and Heyman,H. (2015) LOCUST: An open source load testing tool. http://www.locust.io.
18. Rocco,M. (2015) Celery Task Queue. http://www.celeryproject.org.
19. Pivotal. (2015) RabbitMQ. http://www.rabbitmq.com.
20. Neves,A.R., Ramos,A., Costa,H., Swam,I.I.v., Hugenholtz,J., Kleerebezem,M., Vos,W.d. and Santos,H. (2002) Effect of different NADH oxidase levels on glucose metabolism by *Lactococcus lactis*: Kinetics of intracellular metabolite pools determined by in vivo nuclear magnetic resonance. *Appl. Environ. Microbiol.*, **68**, 6332–6342.
21. Neves,A.R., Ramos,A., Nunes,M.C., Kleerebezem,M., Hugenholtz,J., Vos,W.d., Almeida,J. and Santos,H. (1999) In vivo nuclear magnetic resonance studies of glycolytic kinetics in *Lactococcus lactis*. *Biotechnol. Bioeng.*, **64**, 200–212.
22. Neves,A.R., Ventura,R., Mansour,N., Shearman,C., Gasson,M.J., Maycock,C., Ramos,A. and Santos,H. (2002) Is the glycolytic flux in *Lactococcus lactis* primarily controlled by the redox charge? *J. Biol. Chem.*, **277**, 28088–28098.
23. Sriyudthsak,K., Iwata,M., Hirai,M.Y. and Shiraishi,F. (2014) PENDISC: A simple method for constructing a mathematical model from time-series data of metabolite concentrations. *Bull. Math. Biol.*, **76**, 1333–1351.
24. Zou,C.L. and Feng,J.F. (2009) Granger causality vs. dynamic Bayesian network inference: a comparative study. *BMC Bioinformatics*, **10**, 122.
25. Granger,C.W.J. (1969) Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, **37**, 424–438.
26. Hagberg,A.A., Schult,D.A. and Swart,P.J. (2008) In: Gael,Varoquaux TV and Millman,J (eds). *Proceedings of the 7th Python in Science Conference (SciPy2008)*. Pasadena, pp. 11–15.
27. Sriyudthsak,K., Shiraishi,F. and Hirai,M.Y. (2013) Identification of a metabolic reaction network from time-series data of metabolite concentrations. *PLoS One*, **8**, 1–9.
28. Savageau,M.A. (1969) Biochemical systems analysis: I. Some mathematical properties of rate law for component enzymatic reactions. *J. Theor. Biol.*, **25**, 365-369.
29. Press,W.H., Teukolsky,S.A., Vetterling,W.T. and Flannery,B.P. (2002) *Numerical recipes in C: The art of scientific computing*. 2nd edn. Cambridge University Press, NY.
30. Sriyudthsak,K. and Shiraishi,F. (2010) Investigation of the performance of fermentation processes using a mathmatical model including effects of metabolic bottleneck and toxic product on cells. *Math. Biosci.*, **228**, 1–9.
31. Sriyudthsak,K., Sawada,Y., Chiba,Y., Yamashita,Y., Kanaya,S., Onouchi,H., Fujiwara,T., Naito,S., Voit,E.O., Shiraishi,F. *et al.* (2014) A U-system approach for predicting metabolic behaviors and responses based on an alleged metabolic reaction network. *BMC Syst. Biol.*, **8**, S4.
32. Hairer,E., Nørsett,S.P., Wanner,G., Graham,R.L., Stoer,J. and Varga,R. (1993) *Solving ordinary differential equations I Nonstiff problems*. 2nd edn. Springer-Verlag, Berlin.
33. Hindmarsh,A.C. (1983) In: Al,RSSE (ed). *IMACS Transactions on Scientific Computation*. North-Holland, Vol. **1**, pp. 55–64.
34. Bornstein,B.J., Keating,S.M., Jouraku,A. and Hucka,M. (2008) LibSBML: an API library for SBML. *Bioinformatics*, **24**, 880–881.
35. Sriyudthsak,K. and Shiraishi,F. (2010) Selection of best indicators for ranking and determination of bottleneck enzymes in metabolic reaction systems. *Ind. Eng. Chem. Res.*, **49**, 9738–9742.
36. Dickinson,R.P. and Gelinas,R.J. (1976) Sensitivity analysis of ordinary differential equation systems - A direct method. *J. Comput. Phys.*, **21**, 123–143.
37. SymPy Development Team. (2014) SymPy: Python library for symbolic mathematics. http://www.sympy.org.
38. Hoefnagel,M.H.N., Starrenburg,M.J.C., Martens,D.E., Hugenholtz,J., Kleerebezem,M., Swam,I.I.v., Bongers,R., Westerhoff,H.V. and Snoep,J.L. (2002) Metabolic engineering of lactic acid bacteria, the combined approach: kinetic modelling, metabolic control and experimental analysis. *Microbiology*, **148**, 1003–1013.