BIOINFORMATICS APPLICATIONS NOTE Vol. 21 no. 4 2005, pages 560–561
doi:10.1093/bioinformatics/bti046

Modelling cellular systems with PySCeS

Brett G. Olivier, Johann M. Rohwer and Jan-Hendrik S. Hofmeyr*

Department of Biochemistry, Triple-J Group for Molecular Cell Physiology, University of Stellenbosch, Private Bag X1, Matieland 7602, South Africa

Received on August 13, 2004; revised and accepted on September 23, 2004
Advance Access publication September 28, 2004

ABSTRACT

Summary: The Python Simulator for Cellular Systems (PySCeS) is an extendable research tool for the numerical analysis and investigation of cellular systems.

Availability: PySCeS is distributed as Open Source Software under the GNU General Public Licence and is available for download from http://pysces.sourceforge.net

Contact: bgoli@sun.ac.za

1 INTRODUCTION

Computer modelling has become an integral tool in the analysis and understanding of the reaction networks that underlie cellular processes. Programs such as Jarnac (Sauro, 2000), Gepasi (Mendes, 1997) and ScrumPy (http://bms-mudshark.brookes.ac.uk/ScrumPy/) allow us to simulate the behaviour of these reaction networks; each has its advantages and disadvantages. In this paper we describe the Python Simulator for Cellular Systems (PySCeS), a new open source modelling tool developed by our group. PySCeS is extremely flexible and user-extensible, and is being actively developed and maintained.

2 IMPLEMENTATION

PySCeS is a console-based application written in the Python (http://www.python.org) programming language that runs on both Microsoft Windows (2000/XP) and Linux. It runs as either a single user application on Windows or a multi-user application under Linux. PySCeS makes use of the Scientific Libraries for Python (SciPy), a large collection of mathematical algorithms for science and engineering applications (http://www.scipy.org).

Some of the advantages of using Python with SciPy are as follows. Python is scripted, with all the object-oriented features of a modern programming language, and includes automatic memory management and garbage collection. It is well suited to act as a glue for applications that interface with libraries compiled in other languages, such as C, C++ and Fortran. Both Python and SciPy are freely available and run on a wide variety of operating systems.

*To whom correspondence should be addressed.

It is, of course, possible to build models directly using only Python with SciPy (Olivier *et al*., 2002). Although flexible, this approach does require considerable skill in both numerical analysis and computer programming. PySCeS has been developed to provide a high-level modelling interface that utilizes and extends the low-level capabilities provided by Python and SciPy, making it unnecessary for the modeller to work with advanced programming techniques or low-level numerical algorithms.

Once a working Python/SciPy environment is available (installation details for various operating systems are available from the SciPy website), PySCeS may be installed using the standard Python distribution utilities.

3 USING PySCeS

The first step in the modelling process is to create a model object from an input file. Further information regarding the input file syntax, which describes a model in terms of its stoichiometry and rate equations, can be found in the *Input File Guide* available from the main PySCeS website.

When the model object is first instantiated, its structural properties (Reder, 1988) are determined by LU factorization and calculation of the kernel (K) and link (L) matrices. These matrices are available as model attributes that can be used in further calculations. The differential equations describing the model are automatically generated from the model's rate equations and stoichiometry. Additionally, elementary flux modes can be calculated by way of an interface to MetaTool (Schuster and Hilgetag, 1994; Pfeiffer *et al*., 1999).

Once the input file is loaded, the model is parsed and instantiated as a Python object that can be used for further analysis. Examples of such types of analyses include:

- Time-course simulation using the LSODA solver of ordinary differential equations (Hindmarsh, 1983), which calculates transient changes in the system's variable concentrations and reaction rates.
- PySCeS also includes interfaces to established non-linear solvers, including HYBRD (Garbow *et al*., 1980)

and NLEQ2 (Deuflhard, 2004), which allow the steady-state fluxes and concentrations of the system to be calculated directly.

- Once a steady state has been determined a metabolic control analysis can be performed on the system (Kacser *et al*., 1995; Heinrich and Rapoport, 1974). This includes calculating the elasticities towards the variable metabolites by algebraic or numerical differentiation of the rate equations, as well as the flux and concentration control coefficients by means of matrix inversion (Hofmeyr, 2001).
- The stability of a system can be investigated by the calculation of the eigenvalues. Both stable and unstable steady states can be investigated with continuation techniques [PySCeS provides an interface to the PITCON algorithm (Rheinboldt and Burkardt, 1983)].
- Data and results generated by PySCeS, can either be plotted via the SciPy GnuPlot interface or saved in text, LaTeX or web-ready HTML format. This makes the data easily accessible to other applications or viewable using a web browser. In addition, model properties such as the steady-state solution, control coefficients and structural matrices (e.g. K- and L-matrices) can easily be displayed or written to file.

Although PySCeS has high-level methods that run any of these analyses with a single command, the underlying, low-level methods remain directly accessible. Many of the parameters that control the behaviour of the numerical algorithms (HYBRD, NLEQ2, etc.) can also be customized for specific problems.

More information on both installing and using PySCeS can be found in the *Quick Guide to PySCeS*, available on-line at http://pysces.sourceforge.net

4 CONCLUSIONS

We have found PySCeS to be an extremely useful research tool for understanding and studying cellular systems. The strategy of simultaneously providing access to high-level, easy-to-use modelling modules as well as low-level numerical routines has already proved useful in both research and teaching contexts. We believe that the versatile and extensible nature of PySCeS will facilitate the development of novel ways of modelling and, ultimately, in understanding complex cellular functions.

ACKNOWLEDGEMENTS

This research was partially funded by the South African National Research Foundation.

REFERENCES

Deuflhard,P. (2004) *Newton Methods for Nonlinear Problems*. Springer Series in Computational Mathematics, Springer-Verlag, NY.

Garbow,B.S., Hillstrom,K.E. and More,J.J. (1980) Implementation guide for MINPACK-1. *ANL-80-68*, Argonne National Laboratory.

Heinrich,R. and Rapoport,T.A. (1974) A linear steady-state treatment of enzymatic chains: general properties, control and effector strength. *Eur. J. Biochem.*, **42**, 89–95.

Hindmarsh,A.C. (1983) ODEPACK, a systematized collection of ODE solvers. In Stepleman,R.S. (ed.), *Scientific Computing*. North-Holland, Amsterdam, pp. 55–64.

Hofmeyr,J.-H.S. (2001) Metabolic control analysis in a nutshell. In Yi,T.-M., Hucka,M., Morohashi,M. and Kitano,H. (eds), *Proceedings of the 2nd International Conference on Systems Biology*. Omnipress, Madison, WI, pp. 291–300.

Kacser,H., Burns,J.A. and Fell,D.A. (1995) The control of flux: 21 years on. *Biochem. Soc. Trans.*, **23**, 341–366.

Mendes,P. (1997) Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3. *Trends Biochem. Sci.*, **9**, 361–363.

Olivier,B.G., Rohwer,J.M. and Hofmeyr,J.-H.S. (2002) Modelling cellular processes with Python and SciPy. *Mol. Biol. Rep.*, **29**, 249–254.

Pfeiffer,T., Sanchez-Valdenebro,I., Nuno,J.C., Montero,F. and Schuster,S. (1999) METATOOL: for studying metabolic networks. *Bioinformatics*, **15**, 251–257.

Reder,C. (1988) Metabolic control theory: a structural approach. *J. Theoret. Biol.*, **135**, 175–201.

Rheinboldt,W. and Burkardt,J. (1983) A locally parameterized continuation process. *ACM Trans. Math. Softw.*, **9**, 215–235.

Sauro,H.M. (2000) JARNAC: a system for interactive metabolic analysis. In Hofmeyr,J.-H.S., Rohwer,J.M. and Snoep,J.L. (eds), *Animating the Cellular Map: Proceedings of the 9th International Meeting on BioThermoKinetics*. Stellenbosch University Press, Stellenbosch, South Africa, pp. 221–228.

Schuster,S. and Hilgetag,C. (1994) On elementary flux modes in biochemical reaction systems at steady state. *J. Biol. Syst.*, **2**, 165–182.