

Python, Systems Biology and PySCeS

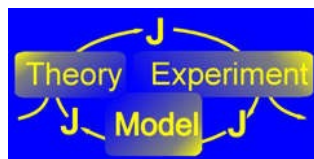
Johann Rohwer

Brett Olivier

Jannie Hofmeyr

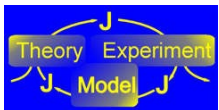
Triple-J Group for Molec. Cell Physiology
Stellenbosch University
South Africa

Max Planck Institute of
Molecular Plant Physiology
Golm



Talk structure

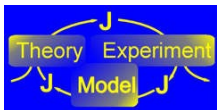
- So what is “Systems Biology”?
- Python in Systems Biology
- PySCeS
- Examples
 - Regulation in a 4-step pathway
 - Modelling sugarcane metabolism
- Conclusion and future prospects



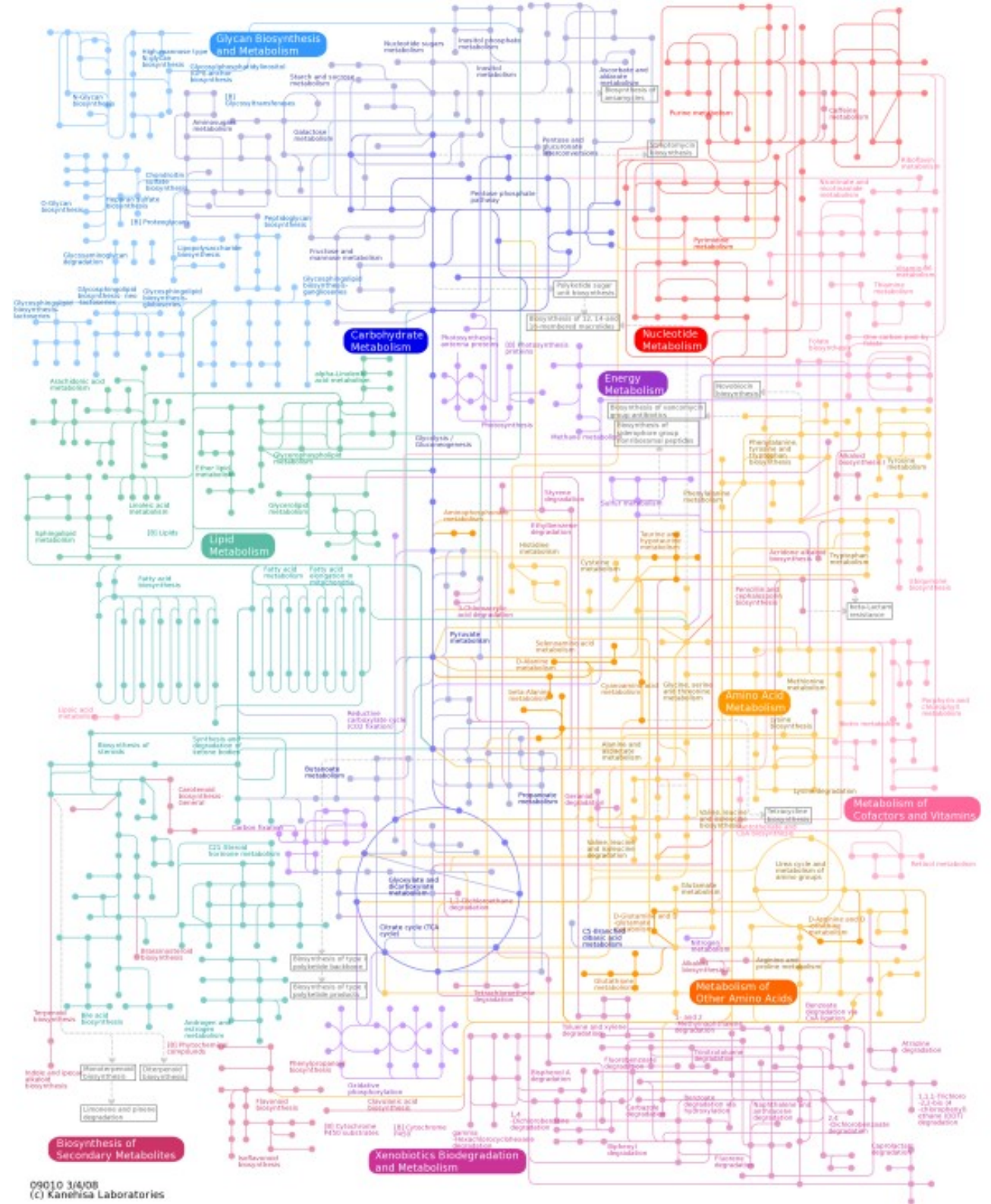
So what is “Systems Biology”?

- Computational Systems Biology
 - ... aims to develop and use efficient algorithms, data structures and communication tools to orchestrate the integration of large quantities of biological data with the goal of modelling ... to create accurate real-time models of a system's response to environmental and internal stimuli.

http://en.wikipedia.org/wiki/Computational_systems_biology



Metabolism



<http://www.genome.jp/kegg/atlas/>

Systems biology – kinetic model

- Differential equation based:

$$\frac{d[\text{fructose}]}{dt} = v_1 + v_2 - v_3$$

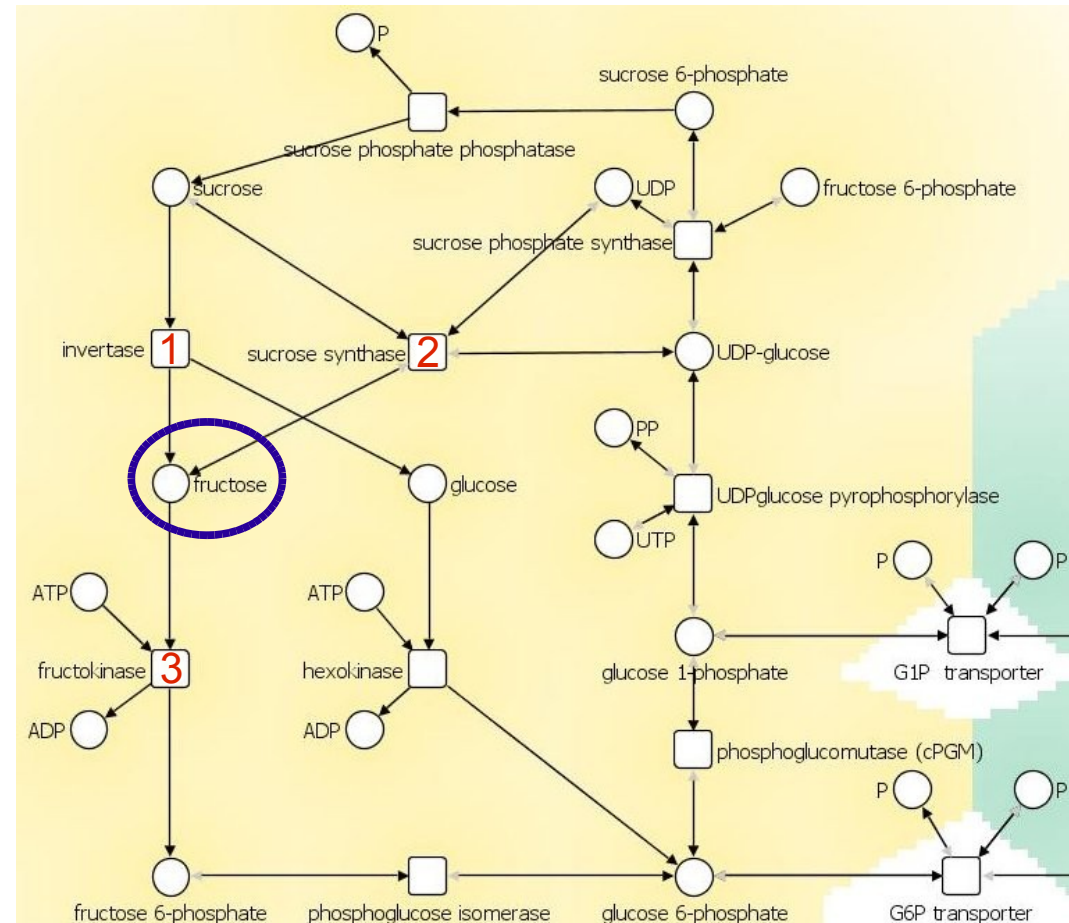
- Set up for all molecular species in system:

$$\frac{d\textcircled{S}}{dt} = \textcircled{N} \textcircled{v}$$

vector of species concentrations

stoichiometric matrix

vector of reaction rates



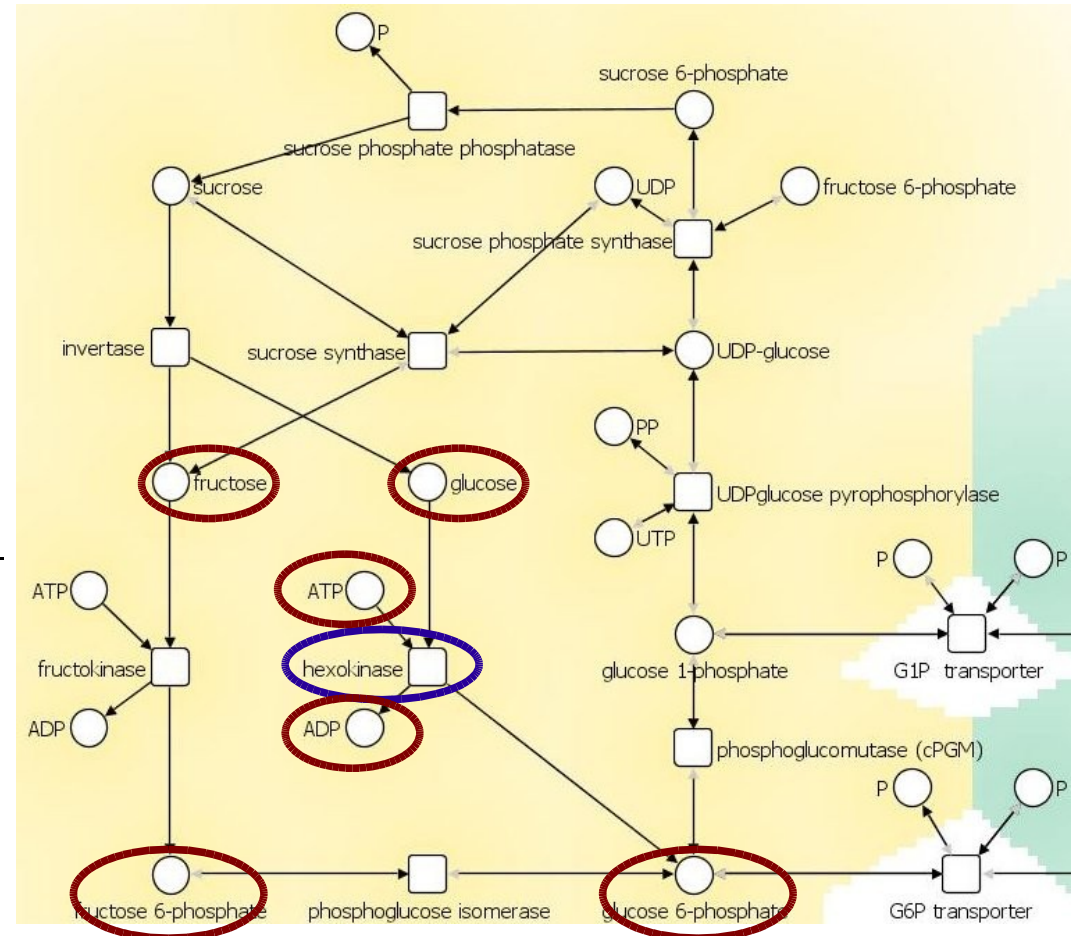
<http://metacrop.ipk-gatersleben.de>

Systems biology – rate equations

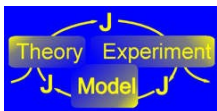
- Enzyme-kinetic rate law for all reactions
- E.g. hexokinase

$$v_{hk} = \frac{V_{max\ hk} \cdot \frac{[Glc]}{K_{m\ Glc}} \cdot \frac{[ATP]}{K_{m\ ATP}}}{\left(1 + \frac{[ATP]}{K_{m\ ATP}}\right) \left(1 + \frac{[Glc]}{K_{m\ Glc}} + \frac{[Fru]}{K_{m\ Fru}} + \frac{[Glc6P]}{K_{i\ Glc6P}} + \frac{[Fru6P]}{K_{i\ Fru6P}}\right)}$$

- Highly nonlinear

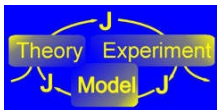


<http://metacrop.ipk-gatersleben.de>



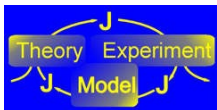
Systems biology – analysis methods

- Time-dependent evolution of system
 - numerically integrate $\frac{dS}{dt} = N \cdot v$ (e.g. LSODA)
 - initial condition S_0
 - trajectory $(t, S(t))$
- Steady-state analysis
 - numerically solve $\frac{dS}{dt} = N \cdot v = 0$ (e.g. HYBRD)
- Control analysis (sensitivity analysis)
 - $\frac{\partial y}{\partial p}$ where y = state variable vector; p = parameter vector
- Stability analysis

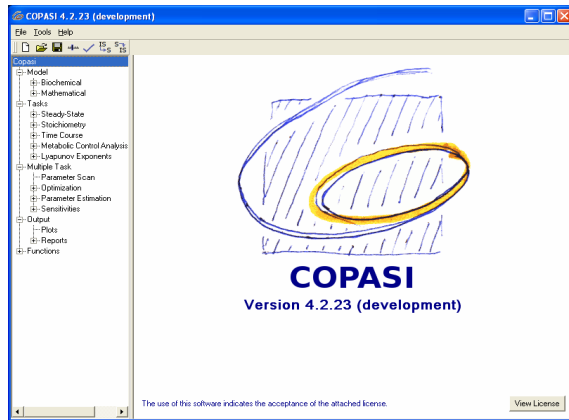


Python and Systems Biology

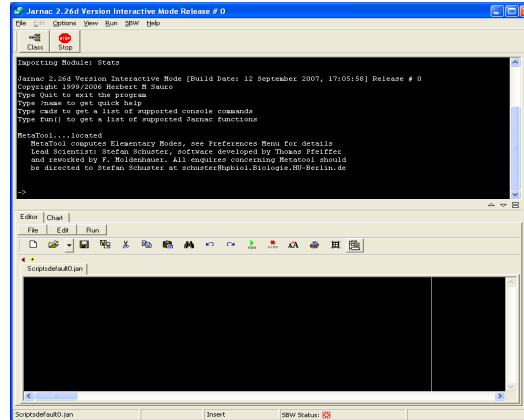
- Computational SysBio relies on numerical analysis
- Availability of NumPy/SciPy makes Python stand **head and shoulders above** other glue languages (e.g. Perl)
 - Heavy reliance on **existing Fortran libraries** (LAPACK, MINPACK)
- Other SysBio software also has Python interfaces
 - libSBML (<http://sbml.org/Software/libSBML>)
 - SBW (<http://sbw.sourceforge.net>)
 - PySBML (<http://pysbml.googlecode.com>)
 - SloppyCell (<http://sloppycell.sourceforge.net>)
 - ScrumPy (<http://mudshark.brookes.ac.uk>)



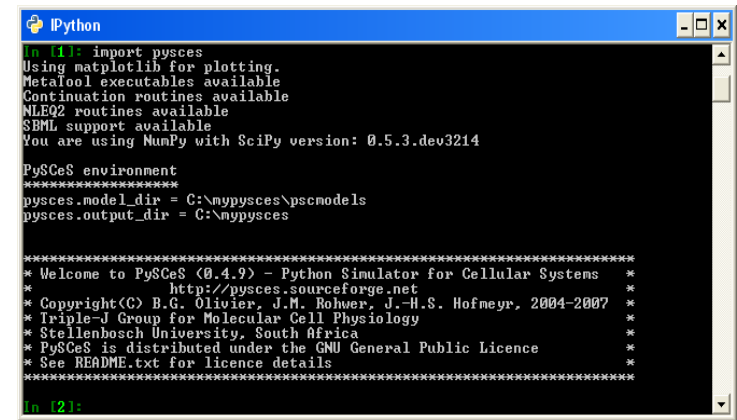
PySCeS, the Python Simulator for Cellular Systems



COPASI



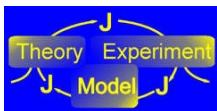
SBW - Jarnac



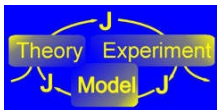
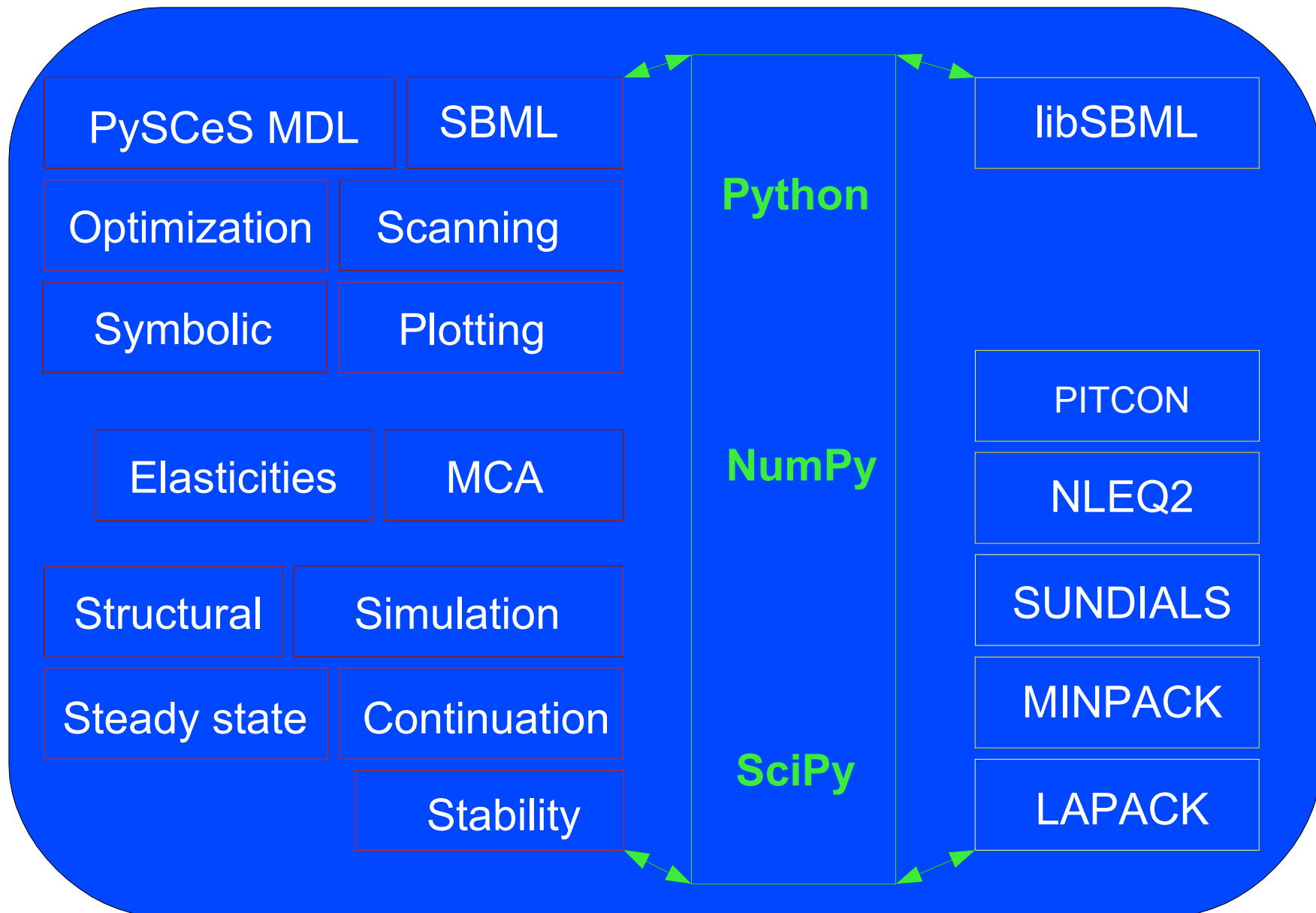
PySCeS

Our design principles:

- **Functionality:** useful to us (and others)
- **Accessibility:** available for a wide range of users in the broader community; not restricted to a specific operating system or hardware platform
- **Flexibility:** software should be extensible allowing for advanced use beyond its initial specifications



PySCeS overview



Model input with the PySCeS MDL

- Parsers developed using David Beazley's PLY

```
Description: PySCeS test model
Species_In_Conc: True
```

```
Compartment: C1, 1.0, 3
Compartment: M1, 1.0, 2
```

```
FIX: x0 x3
```

```
R1@C1: x0 = s0
        C1*(k1*x0 - k2*s0)
```

```
k1 = 7.0
k2 = 1.0
```

```
R2@M1: s0 = x3
        M1*(k3*s0 - k4*x3)
```

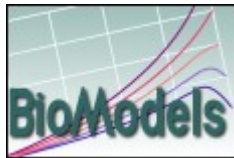
```
k3 = 5.0
k4 = 1.0
```

```
x1@C1 = 10.0
x3@M1 = 1.0
s0@M1 = 1.0
```

Model input: SBML (sbml.org)

- SBML import and export using libSBML and PLY

Model databases



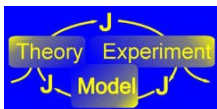
<http://biomodels.net>



<http://jjj.biochem.sun.ac.za>

SBML compatible software

BALSA	DBsolve	Molecuizer	SBMLR
BASIS	Dizzy	Monod	SBMLSim
BIOCHAM	E-CELL	Narrator	SBMLToolbox
BioCharon	ecellJ	NetBuilder	SBLiD
ByoDyn	ESS	Oscill8	SBToolbox
biocyc2SBML	FluxAnalyzer	PANTHER Pathway	SBW
BioGrid	Fluxor	PathArt	SClpath
BioModels	Gepasi	PathScout	Sigmoid*
BioNetGen	Gillespie2	PathwayLab	SigPath
BioPathway Explorer	INSILICO discovery	Pathway Tools	SigTran
Bio Sketch Pad	JACOBIAN	PathwayBuilder	SIMBA
BioSens	Jarnac	PATIKAwab	SimBiology
BioSPICE Dashboard	JDesigner	PaVESy	Simpathica
BioSpreadsheet	JigCell	PET	SimWiz
BioTapestry	JSim	PNK	SloppyCell
BioUML	JWS Online	Reactome	SmartCell
BSTLab	Karyote*	ProcessDB	SRS Pathway Editor
CADLIVE	KEGG2SBML	PROTON	StochSim
CellDesigner	Kinsolver*	pysbml	StochKit
Cellerator	libSBML	PySCeS	STOCKS
CellML2SBML	MathSBML	runSBML	TERANODE Suite
Cellware	MesoRD	SABIO-RK	Trelis
CL-SBML	MetaboLogica	SBML ODE Solver	Virtual Cell
CLEML	MetaFluxNet	SBML-PET	WebCell
COPASI	MMT2	SBMLEditor	WinSCAMP
Cytoscape	Modesto	SBMLmerge	XPPAUT



Instantiating a model

```
In [1]: import pysces
```

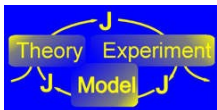
```
*****
* Welcome to PySCeS (0.5.0) - Python Simulator for Cellular Systems *
* http://pysces.sourceforge.net *
* Copyright(C) B.G. Olivier, J.M. Rohwer, J.-H.S. Hofmeyr, 2004-2008 *
* Triple-J Group for Molecular Cell Physiology *
* Stellenbosch University, South Africa *
* PySCeS is distributed under the GNU General Public Licence *
* See README.txt for licence details *
*****
```

```
In [2]: mod = pysces.model('moiety_branch')
```

```
Using model directory: C:\mypysces\pscmmodels
C:\mypysces\pscmmodels\moiety_branch.psc loading .....
```

```
In [3]: mod.doLoad()
```

```
Parsing file: C:\mypysces\pscmmodels\moiety_branch.psc
Calculating L matrix . . . . . done.
Calculating K matrix . . . . . done.
```



PySCeS structural analysis

- Non-linear root finding requires calculation of both left and right **nullspaces**, typically of non-square matrices
- Currently implemented using **LU factorisation** via (**scipy.linalg.(c/f)lapack.dgetrf**), column reordering and Gauss-Jordan elimination

In [4]: `mod.showN()`

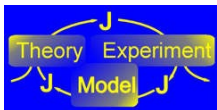
Stoichiometric matrix (N)

	R1	R2	R3	R4
S2	-1.0	1.0	0.0	0.0
S1	1.0	0.0	-1.0	-1.0
S3	1.0	-1.0	0.0	0.0

In [5]: `mod.showNr()`

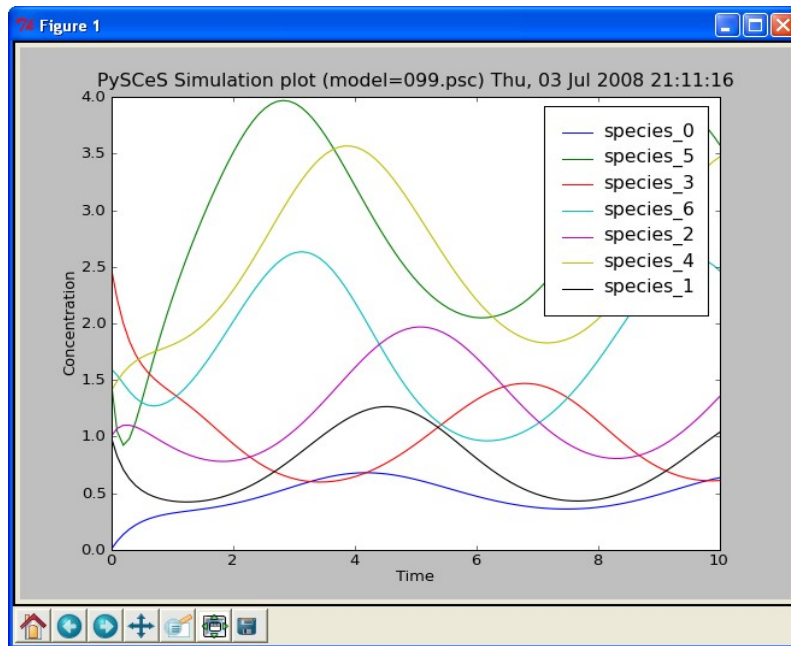
Reduced stoichiometric matrix (Nr)

	R1	R2	R3	R4
S1	1.0	0.0	-1.0	-1.0
S2	-1.0	1.0	0.0	0.0

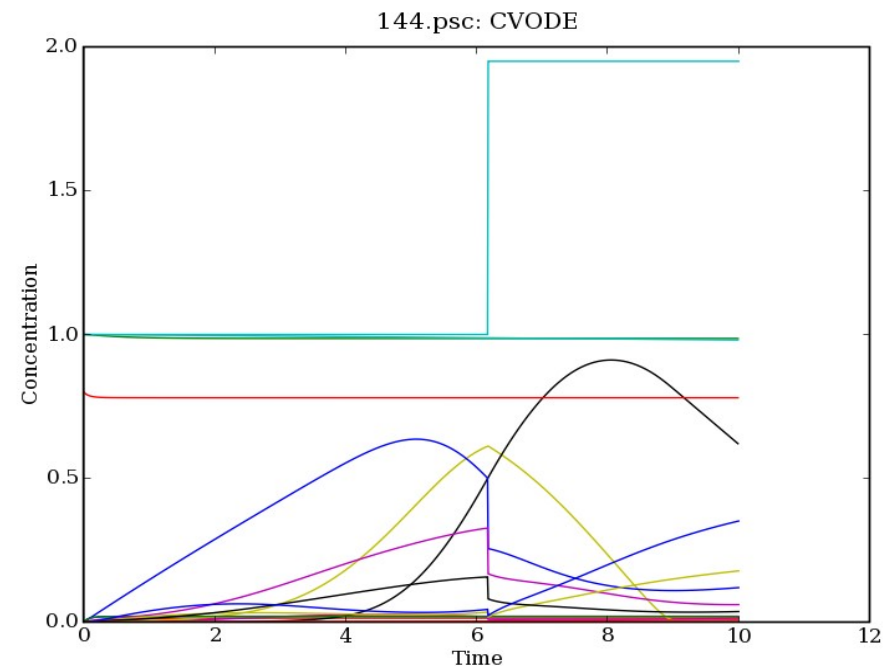


Time course ($dS/dt = N.v$)

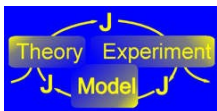
- LSODA fast but ageing (`scipy.integrate.odeint`)
- SUNDIALS CVODE (`ctypes` interface, PySundials, <http://pysundials.sourceforge.net>)
- 2D plotting with `Matplotlib`



LSODA

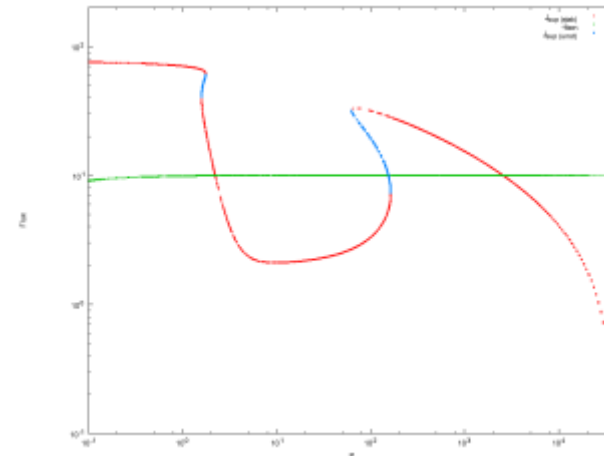
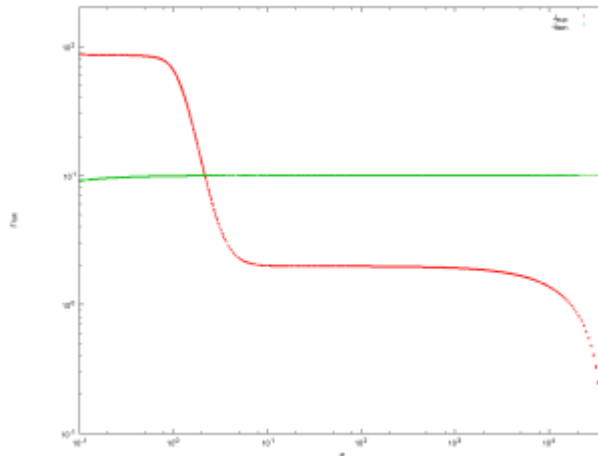


CVODE

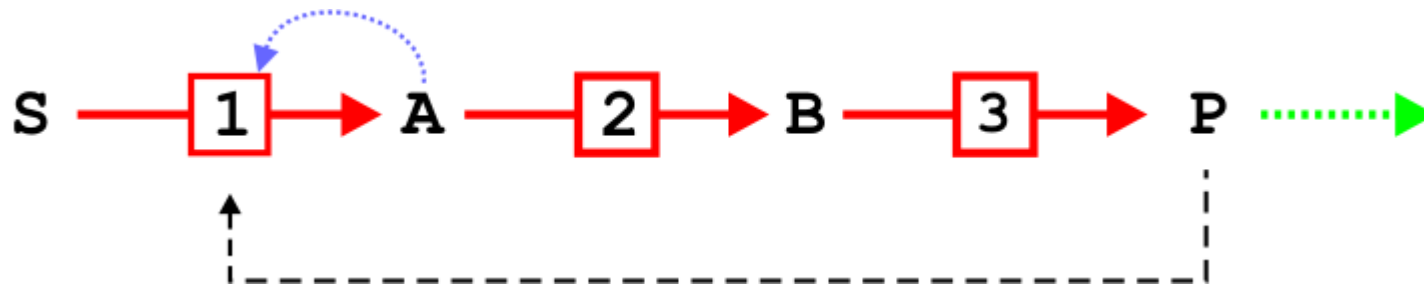


Steady state ($dS/dt = N.v = 0$)

- Important for analysing control and regulation of cellular systems
- Non-linear root finders
 - Direct:
 - HYBRD (SciPy)
 - NLEQ2 (<http://www.zib.de/Numerik/numsoft/ANT/nleq2.en.html>) own interface with f2py
 - KINSOL from SUNDIALS (own interface with ctypes)
 - Integration for long t :
 - LSODA, CVODE
- Direct solvers preferred: parameter scans need millions of solves



Example 1: Cellular regulation with PySCeS/Kraken



A four enzyme linear system with negative feedback

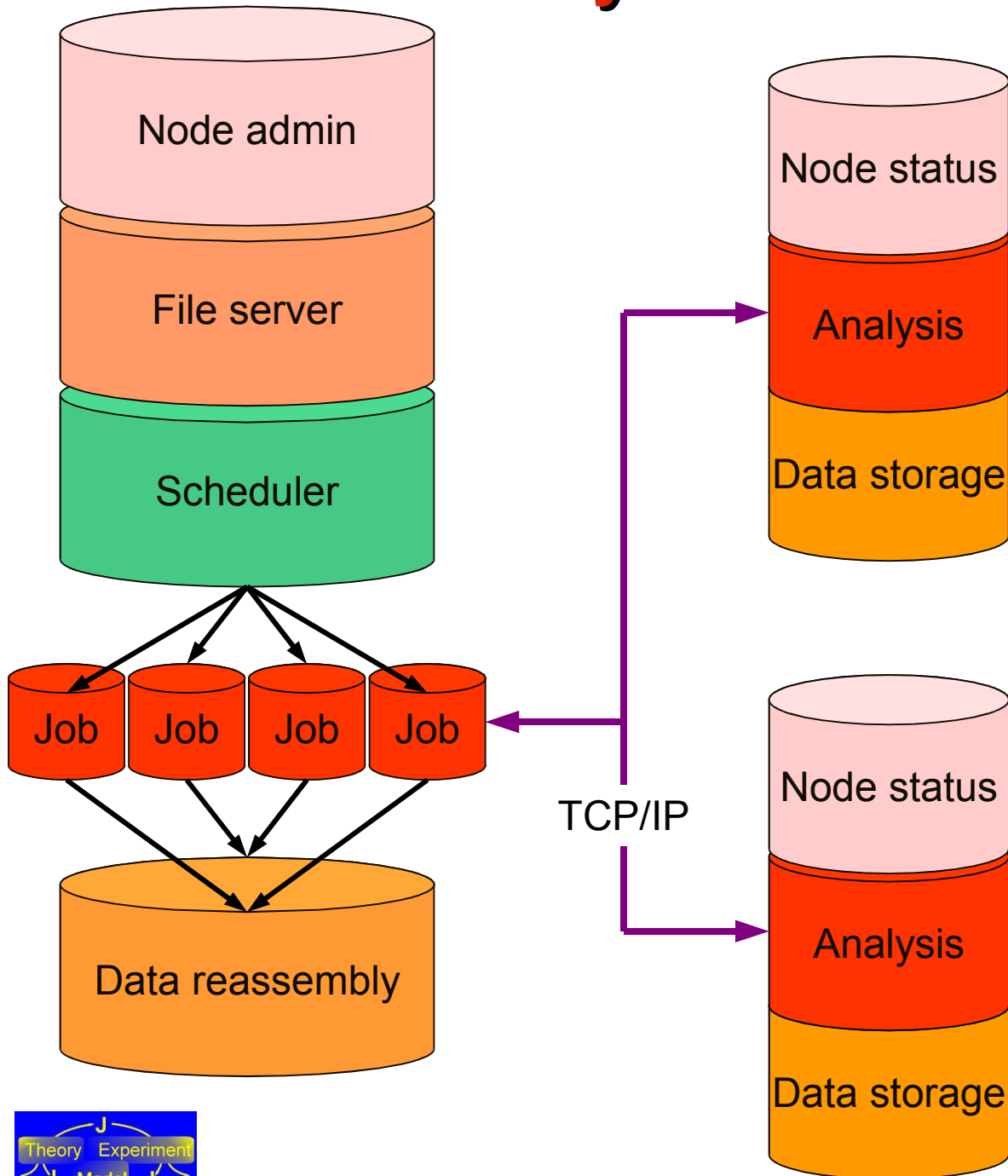
$$v_{\text{Hill}} = V_f \frac{s}{s_{0.5}} \frac{\left[\frac{s}{s_{0.5}} + \frac{a}{a_{0.5}} \right]^{h-1}}{\left[\frac{s}{s_{0.5}} + \frac{a}{a_{0.5}} \right]^h + \frac{1 + \left[\frac{p}{p_{0.5}} \right]^h}{1 + \alpha \left[\frac{p}{p_{0.5}} \right]^h}} \left[1 - \frac{\Gamma}{K_{\text{eq}}} \right]$$

Reaction 1

$$\frac{\frac{V_f}{K_s} \cdot \left(S - \frac{P}{K_{eq}} \right)}{\left(1 + \frac{S}{K_s} + \frac{P}{K_p} \right)}$$

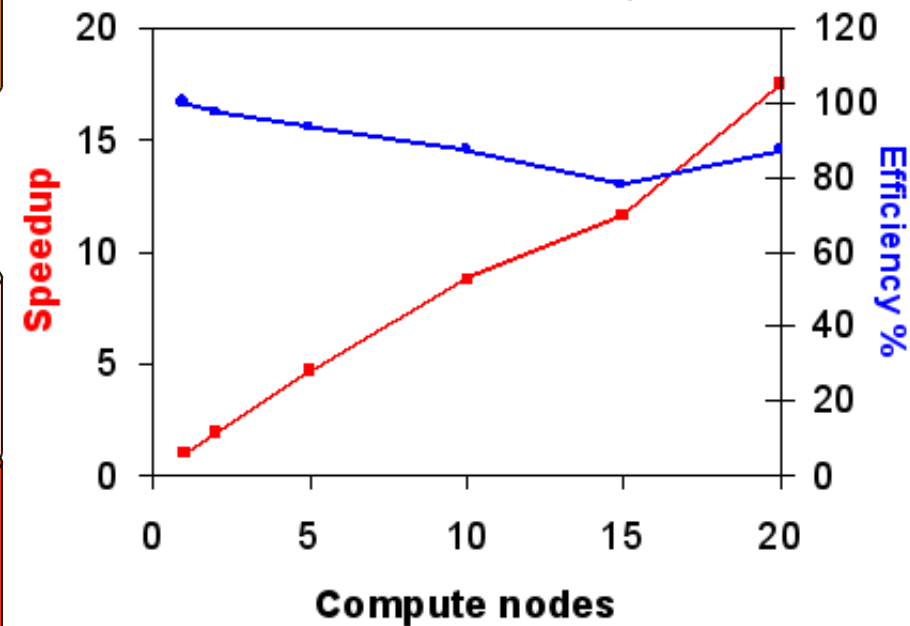
Reactions 2 and 3

PySCeS/Kraken



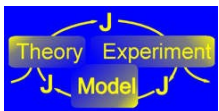
- **Embarrassingly parallel** high level analysis (e.g. parameter scans)
- Only **standard library** components
- **Extensible** design

Performance on a 500000 point scan

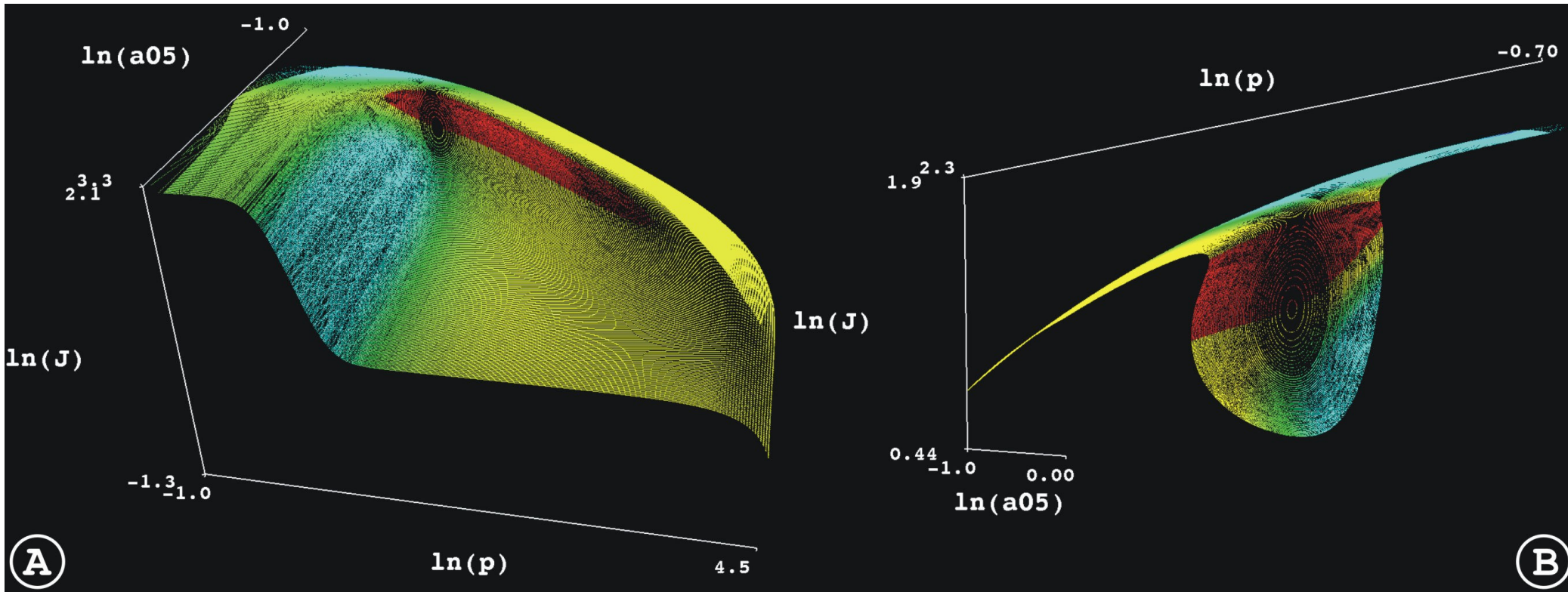


$$\text{speedup} = \frac{\text{time}_{(1 \text{ node})}}{\text{time}_{(n \text{ nodes})}}$$

$$\text{efficiency} = \text{speedup} / \# \text{ of nodes}$$



3D Rate characteristics

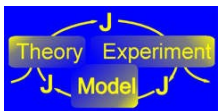


(A) 2.2 million point scan showing the bifurcation topology

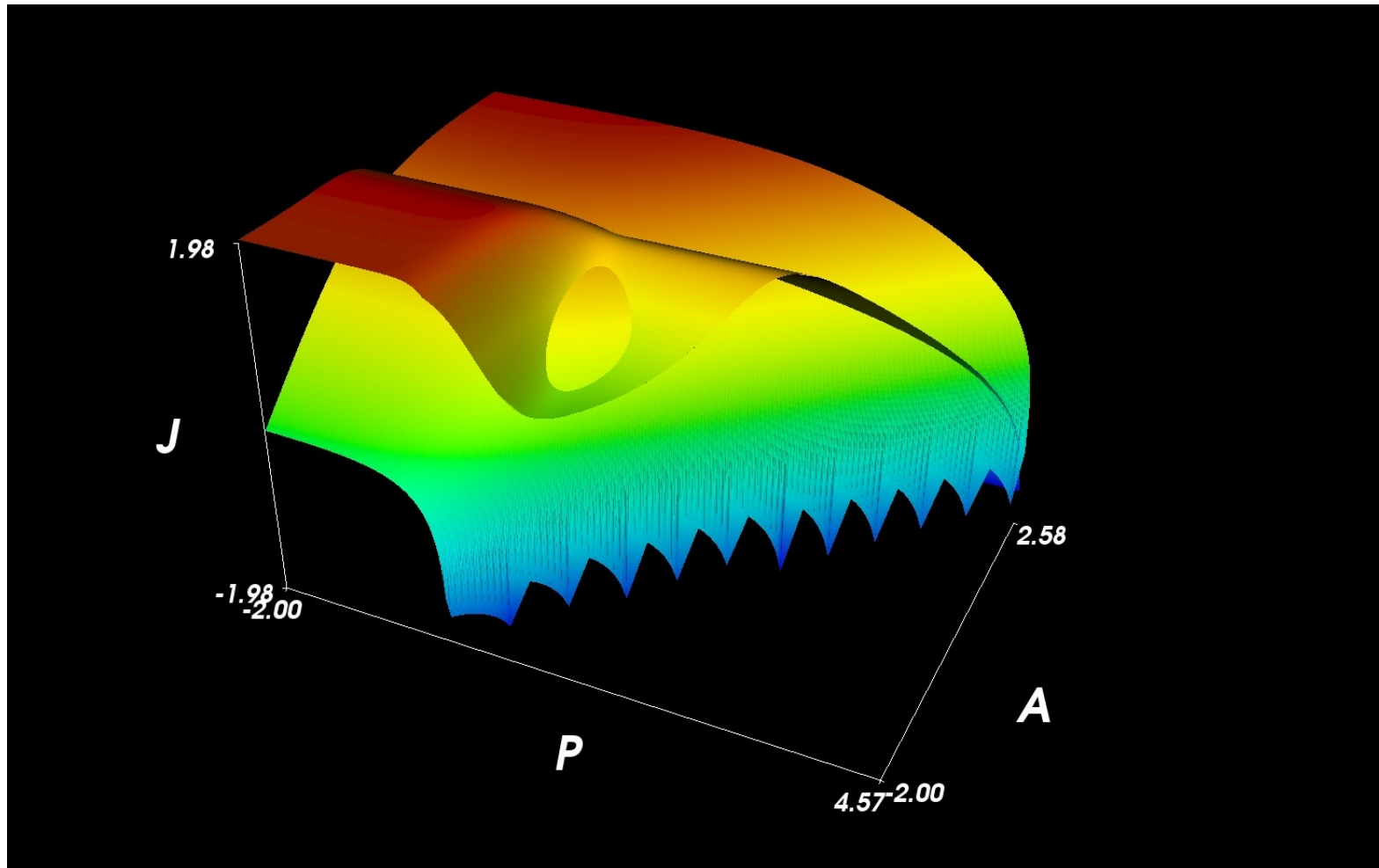
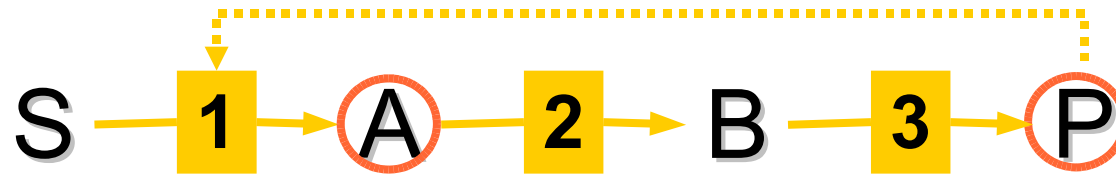
(B) 500000 point magnification profiling the isola and double hysteresis

Red states are unstable (positive Eigenvalues of Jacobian)

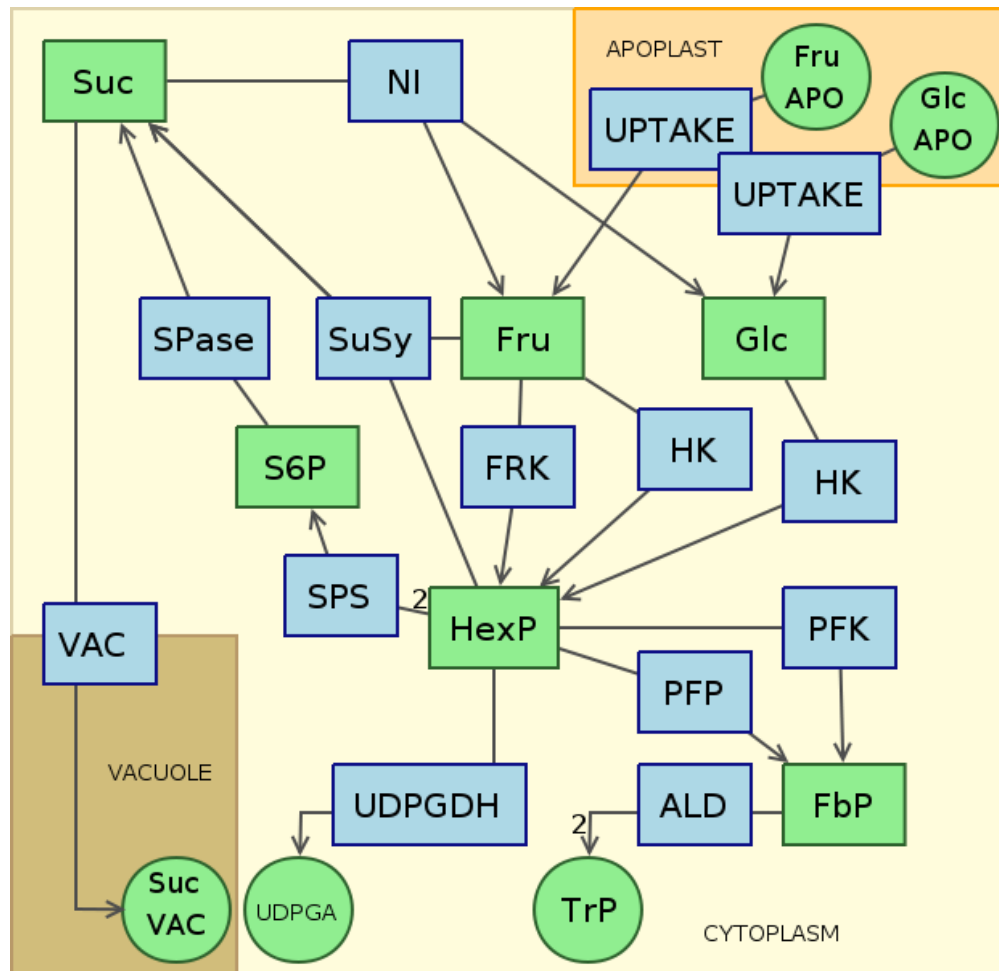
Visualised with [MayaVI2](#) powered by [Enthought.com](#)



Disentangling the instability



Example 2: Sucrose accumulation in sugarcane



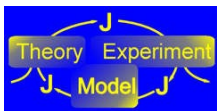
- Sugarcane accumulates large quantities of sugar as the plant matures
- Build model to identify steps controlling this process

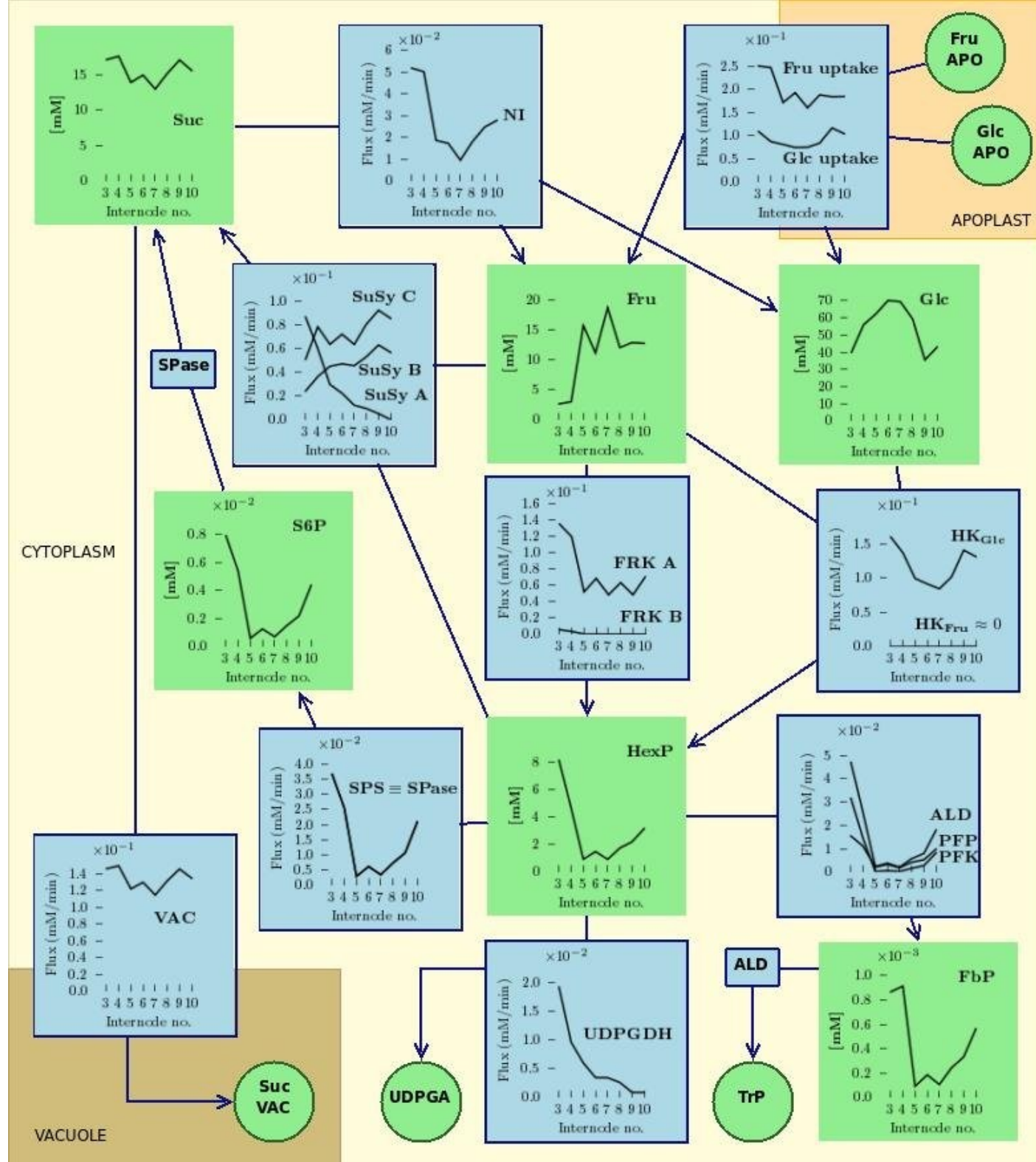
Uys et al. (2007) *Phytochemistry* **68**:2375

Methods



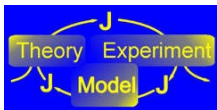
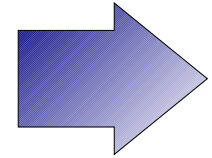
- Metabolism within an internode regarded as constant
- Reference steady-state model of sucrose metabolism
- Internode specific parameters
 - from internode 3 (young) to 10 (mature tissue)



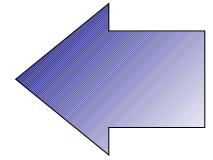


Current and future developments

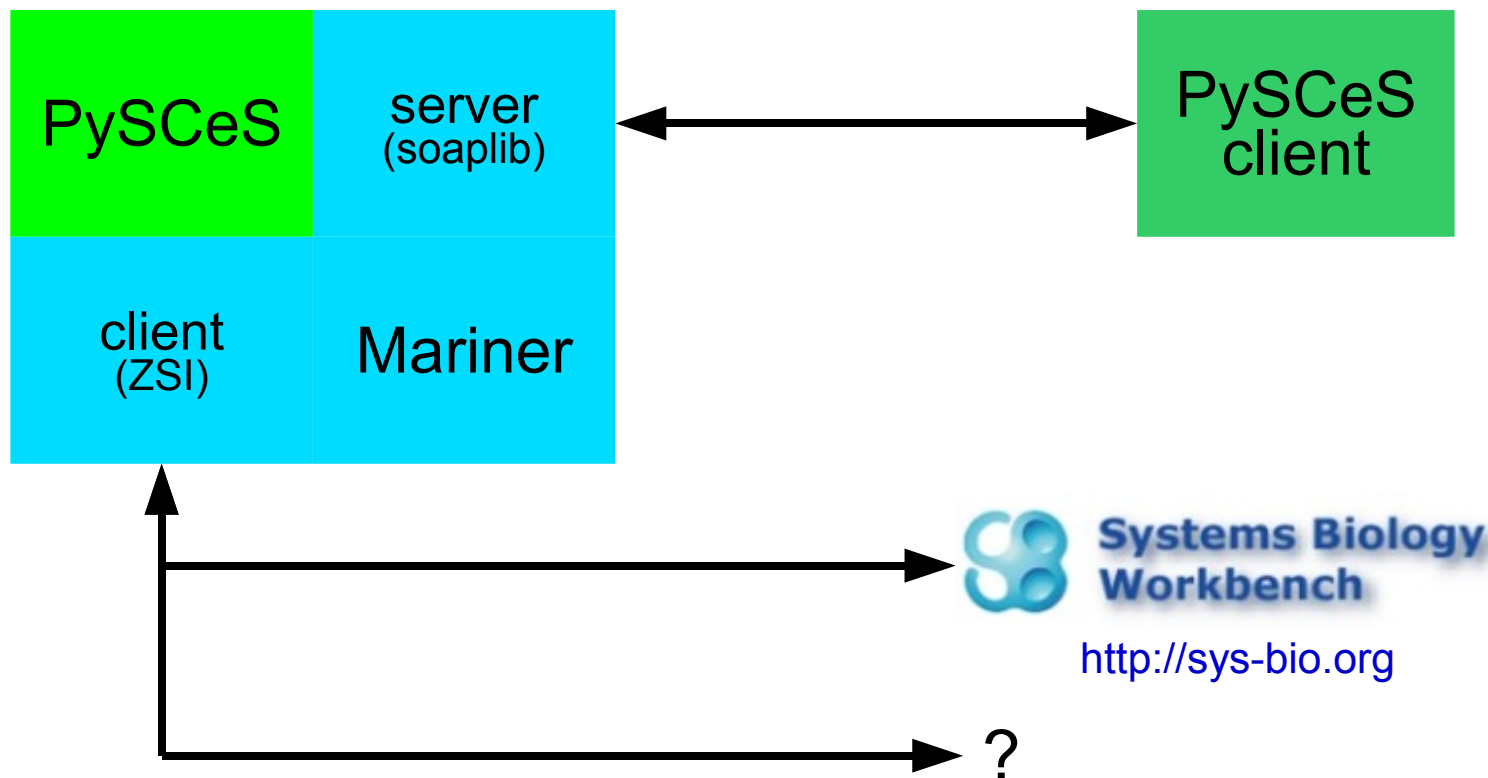
- PySCeS/Mariner
 - web-application framework using **soaplib**
- A better low-res 3D-plotting solution
- Symbolic control analysis (using **Maxima**)
- Parameter fitting and optimisation
 - integration of global and local methods



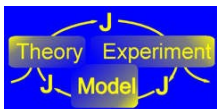
PySCeS/Mariner



- A web services module for PySCeS
 - **consume** other available SOAP-based web resources
 - **expose** PySCeS SOAP services



ZSI: <http://pywebsvcs.sourceforge.net>
soaplib: <http://trac.optio.webfactional.com/wiki/soaplib>



Acknowledgements

- Brett Olivier
(lead developer)
- Jannie Hofmeyr
- Lafras Uys (sugarcane)
- James Dominy
(PySUNDIALS)
- Tim Akhurst
(SymCA)
- Jacky Snoep
(JWS Online)

- National Bioinformatics
Network
(South Africa)



- Alexander von Humboldt
Foundation

