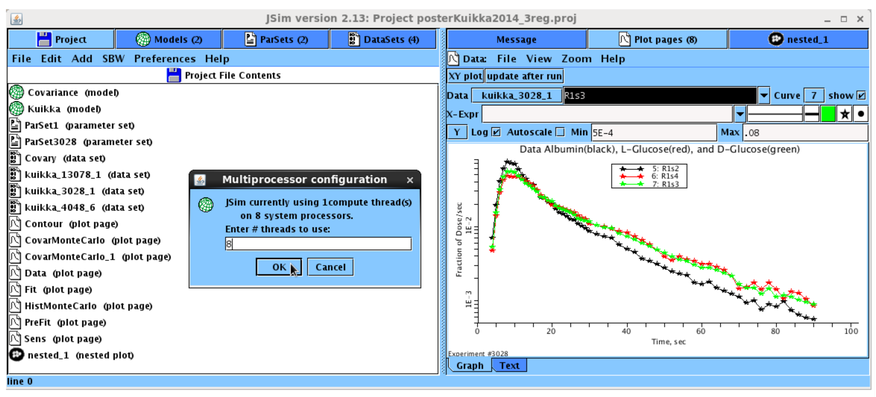
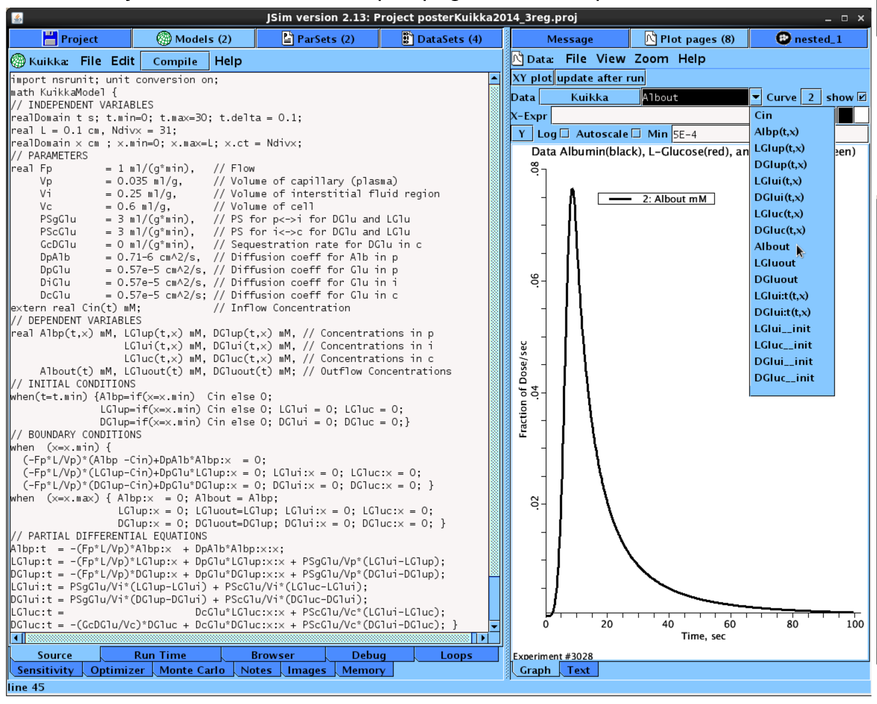
Abstract: JSim is a simulation system for developing models, designing experiments, and evaluating hypotheses on physiological and pharmacological systems through the testing of model solutions against data. It is designed for interactive, iterative manipulation of the model code, different data sets, parameter sets, and for making comparisons among different models running simultaneously or separately. Interactive use is supported by a large collection of graphical user interfaces handling model writing and compilation diagnostics, defining input functions, model runs, selection of algorithms solving ordinary and partial differential equations, run-time multidimensional graphics, parameter optimization (8 methods), sensitivity analysis, and Monte Carlo simulation for defining confidence ranges. JSim uses Mathematical Modeling Language (MML) a declarative syntax specifying ODEs and PDEs. The model coding and displaying of results is basically simply defining the parameters, variables, and writing the equations in MML in usual form (except for using x:t for dx/dt) making it good for teaching modeling as well as for model analysis for research. JSim can be run as a batch job. JSim can automatically translate models from the repositories for Systems Biology Markup Language (SBML) and CellML models. Stochastic modeling is supported. JSim encourages assigning physical units to variables and parameters and automates the first steps of verification testing by checking for dimensional balance with automatic unit scaling included in the computation. The JSim Project File sets a standard for reproducible modeling analysis by including in one ascii file everything for a complete analysis of experiments: the data, the quantitative hypotheses defined by the models, and the data analysis. JSim is open source; it and about 400 human readable open source models are available at http://www.physiome.org/jsim/.

JSim, an open-source modeling system for data analysis and reproducibility in research

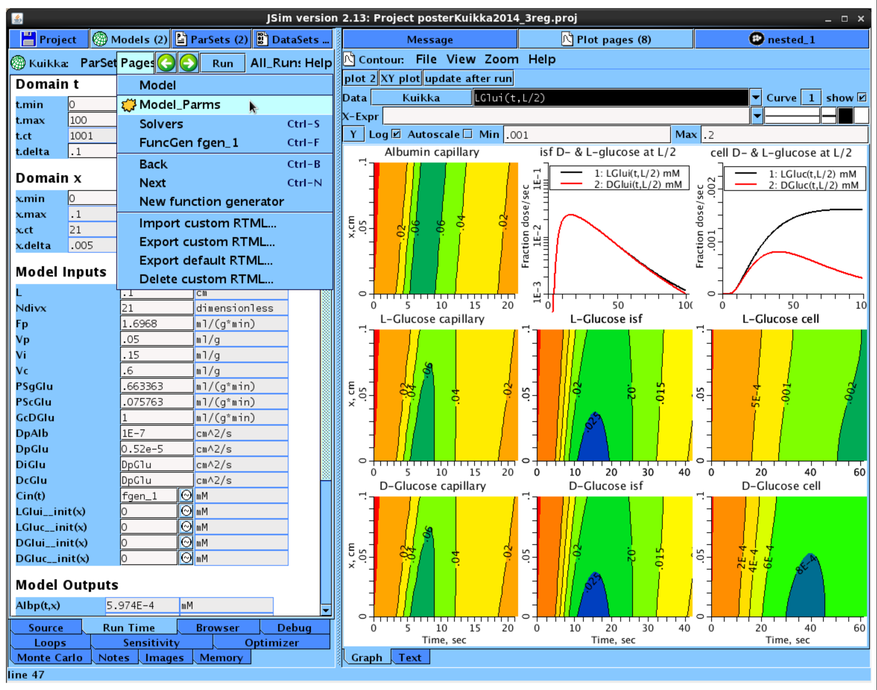
James B. Bassingthwaighte, Erik Butterworth, Bart Jardine, Gary M. Raymond, Maxwell L. Neal
Bioengineering, University of Washington, Seattle, WA 98195-5061



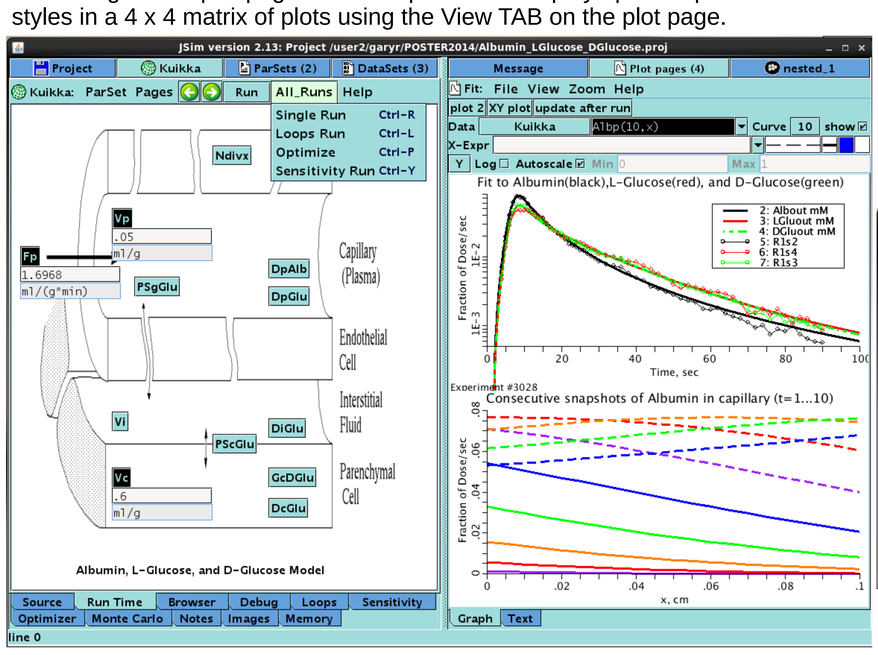
(1) Left: A JSim project contains all the models, parameter sets, data sets, plot pages, and note pages in an organized format. Jsim models can use as many computation threads as the number of system processors to speed computation and runs on Mac, Window, and Linux platforms. Right: Plots are formatted interactively in JSim and saved as plot pages. This is a plot of data.



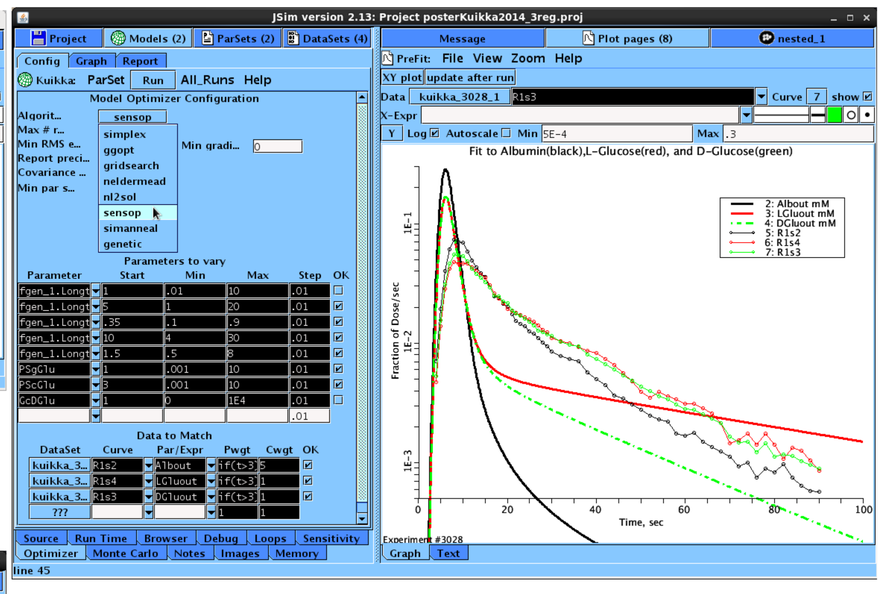
(2) Left: JSim models are written in Mathematical Modeling Language (MML) using declarative syntax (equations). Right: Plots are formatted interactively by using menu selections or entering variable names manually before or after the model is run.



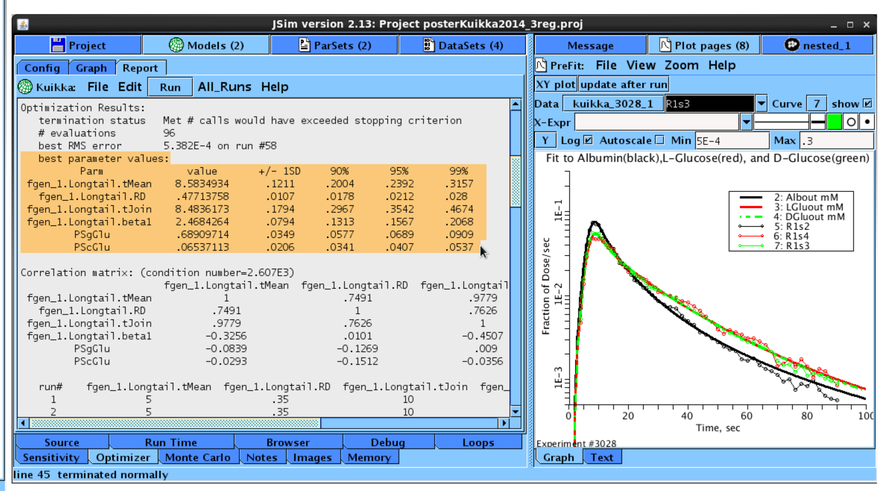
(3) Left: Every model is provided with a similar interface found on the RunTime TAB by selecting the Pages TAB then selecting Model_Parms from the pull down menu. Right: The plot page can be expanded to display up to 16 plots of different styles in a 4 x 4 matrix of plots using the View TAB on the plot page.



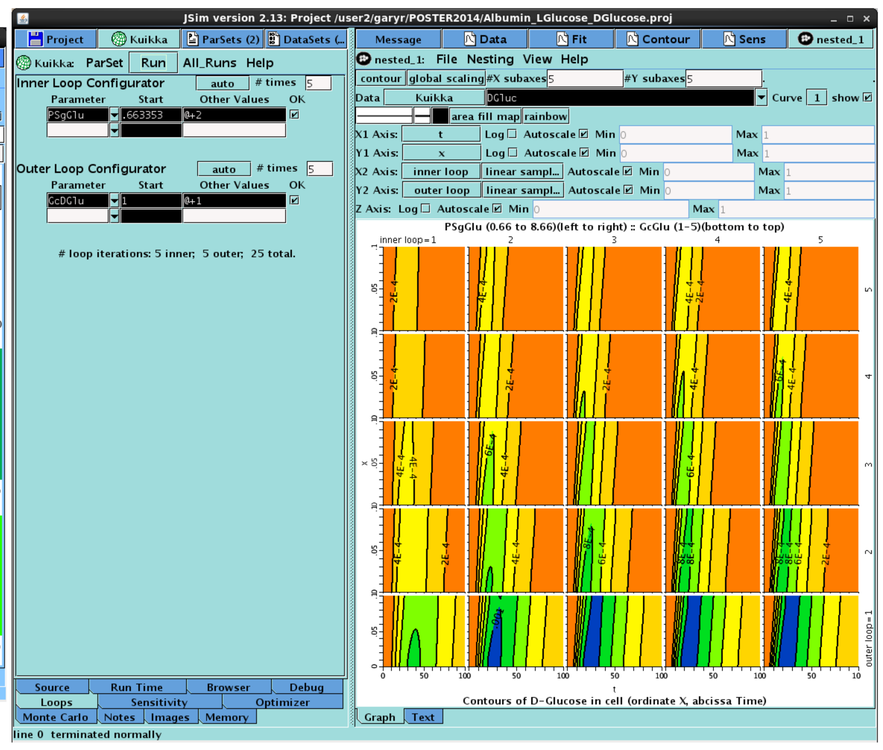
(4) Left: Customized user interfaces are easily generated as an alternative to the standard menu see at thebottom of the previous column. Different kinds of runs are available from the All_Runs TAB. Right: (Upper) Model Fits to the data are displayed. (Lower) Consecutive snapshots at t=1 to 10 seconds in intervals in 1 sec of Albumin in the capillary are captured: 1 sec (solid purple) ... 5 sec (solid Blue), 6 sec (dashed purple) … 10 sec (dashed blue).



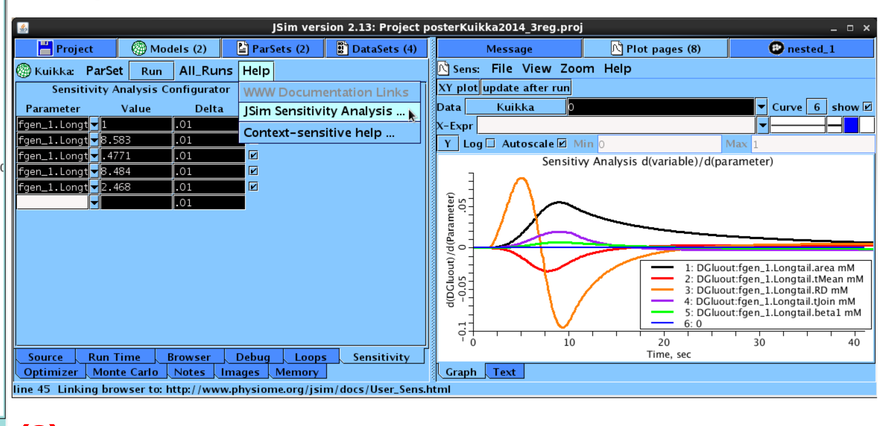
(5) Left: There are eight different optimizers for fitting models to data available using the optimizer Graphical User Interface. This user interface is available by selecting the Config TAB. Right: The data is plotted with the un-optimized parameters.



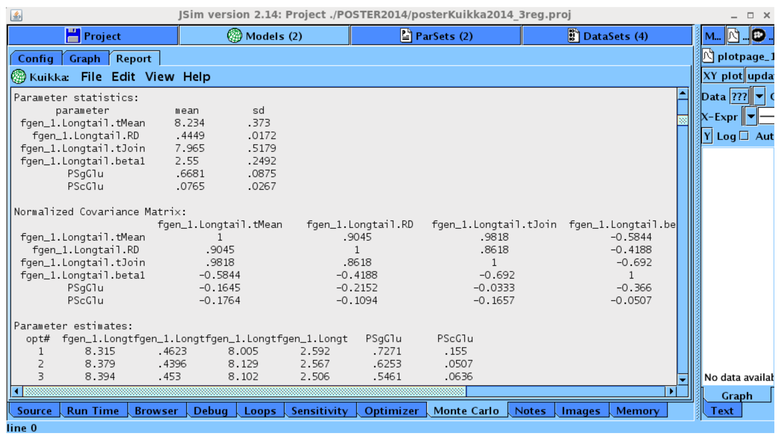
(6) Left: A detailed report from running the optimizer is available under the Report TAB. It includes confidence intervals for each parameter and the correlation between parameters. Right: The data is plotted with the optimized parameters.



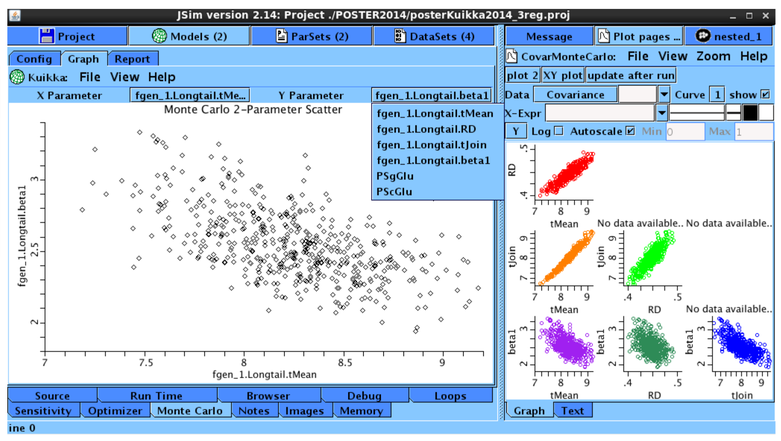
(7) Left: Model's response of changes of parameter values can be analyzed by "looping" on a parameter. If more that one parameter is being changed, the results can be viewed using nested plots. Right: In the 5x5 matrix of plots, PsgGlu, the exchange rate for L- and D-Glucose from the capillary to the interstitial fluid region is varied from from left to right with values 0.66, 2.66, 4.66, 6.66 and 8.66. The consumption rate for D-Glucose in the cell is varied from bottom to top with values of 1, 2, 3, 4, and 5. Each individual plot has the cell length as the vertical axis and time as the horizontal axis.



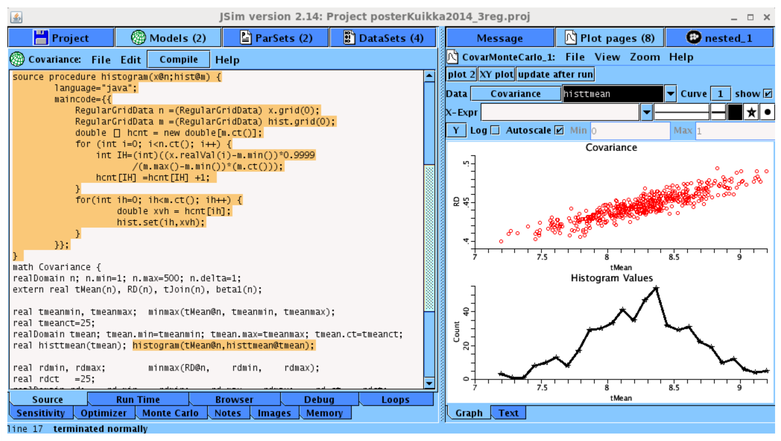
(8) Left: Sensitivity of an model output variable as a result of a small change in parameter value is calculated. In this instance the parameters used to generate a Longtail input function are used to note the change in DGluout. Right: Where the sensitivity function of a parameter is positive (negative) a small positive change in the value of the parameter will cause a positive (negative) change in the variable.



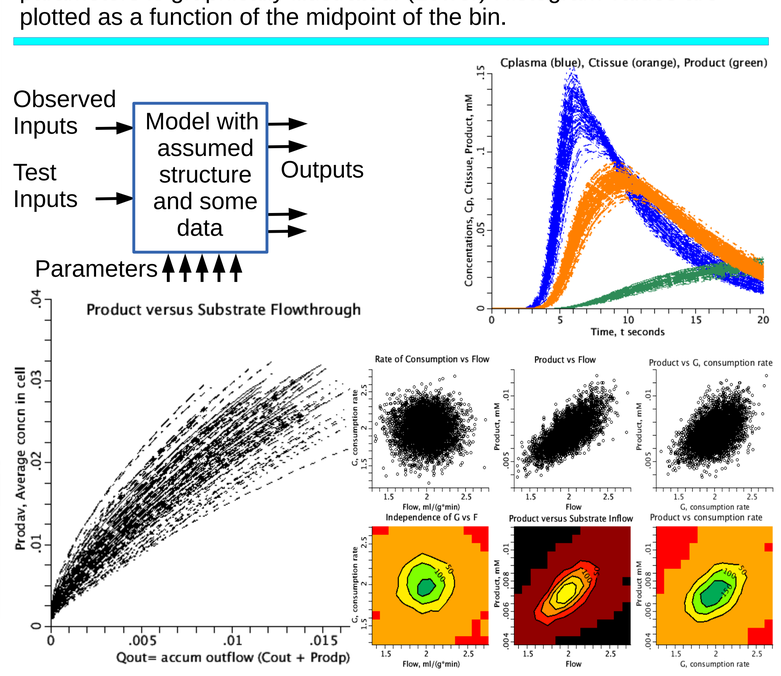
(9) Left: The Monte Carlo routines utilize the values from the optimizer GUI. Noise is added to the data being optimized. The noise can be either uniform [-1,1] or Gaussian (mean 0, standard deviation) and either additive or proportional. The number of cases is set. There are two types of reports generated: a global report as illustrated here and a complete set of optimization reports, one for each case. Right: The GUI's can be resized.



(10) Left: The covariance between two parameters is illustrated by a scatter plot, one of the many plots available from the Graph TAB for the Monte Carlo GUI. Right: The data set of parameters was saved and used by the second model in this project (see next panel below).



(11) (From panel above) Left: The optimized parameters from Monte Carlo analysis can be directly stored as a data set and used by a second model (also stored in this project). This model is written in MML, but also accesses two Java procedures, minmax (finds the minimum and maximum of an array) and histogram (constructs histogram values from an array). Right: (Upper) The covariance of two parameters is graphically illustrated. (Lower) Histogram values are plotted as a function of the midpoint of the bin.



(12) Model Predictions are uncertain because of uncertainties in: (1) inputs to the system, (2) configuration of the model itself, and (3) model parameter values. Product formation (e.g. energy) depends on transport in series: substrate delivery to tissue by Flow, conductance into cells by permeation, and rates of metabolic reactions, G. With constrained random Gaussian parameter values, the solutions show divergent paths, (upper right and lower left). While parameter values are independent of each other (lower right (leftmost panels)) G vs.Flow, product formation depends on Flow (lower right (middle panels)) and consumption rate (lower right (rightmost panels)) non-randomly, with a predictable probability (contour plots).

This software was developed with support from NIH grants HL088516 and HL073598, NIBIB grant BE08417 and the Virtual Physiological Rat program GM094503 (PI: D.A. Beard).