The Systems Biology Workbench (SBW) Version 1.0: Framework and Modules

Introduction
Progress in molecular biotechnology has fuelled an explosion in the development of software tools. Regrettably, developers often end up reinventing similar facilities in separate software packages. In an effort to make it more attractive for developers to share rather than re-implement resources, we have implemented the Systems Biology Workbench (SBW), a free, open-source, application integration environment. Our goal has been to create a framework simple enough that software authors find it easier to provide an SBW interface than to re-create functionality available in other tools. By doing so, we hope developers can concentrate on creating best-of-breed solutions in their areas of expertise.

What Does SBW Provide?
SBW provides libraries for enabling applications to learn about and communicate with each other. The applications may be running on separate computers.

The SBW Broker
The SBW Broker starts applications on demand, and coordinates computing on a given computer. A Broker is started automatically for the user if one is not running when the first SBW application starts.

Features of SBW Version 1.0
- Languages supported: C, C++, Delphi, Java, Perl, Python, and Ruby
- Windows (95, 2000, XP), Linux, and macOS X in the near future.
- Secure, distributed operations via SSH, featuring remote startup of brokers and applications.
- CORBA gateway for bidirectional communication between SBW-based apps and CORBA-based apps.
- Collection of basic applications provided with the SBW distribution, including:
  - A simple stochastic simulator based on the Gillespie-Brownian version of the Gillespie algorithm
  - An SBML-to-MATLAB ODE & Simulink translator
  - An SBML reader tool that allows a program to extract (via an API) components of an SBML model
  - A “clipboard” module that stores an SBML model description, and allows the easy transfer of models between separate modules
  - A “browser” module that allows querying SBW for registered modules and producing descriptions of each module’s interface in Java or CORBA IDL
  - A simple plotting module for time-series data
  - A generic simulation control GUI interface
- A collection of tutorial example modules in C, C++, Delphi and Java
- Extensive documentation—in addition to overview documents and published papers, every language library has its own programmer’s manual and API reference.

SBW in Action: A Sample Session
Here is an example of using several SBW-enabled tools to create and simulate a two-compartment model of a hypothetical single-gene oscillatory circuit.

In this highly simplified model, there is a gene G, which encodes its own repressor and is transcriptionally activated at a constant rate, V. Transcriptional activation of a gene G (which normally involves enzymatic reactions) is summarized here as the production of active RNAP from source material, and degradation to inactive. Transcribed mRNA is then transported from the nucleus into the cytoplasm, where it is translated into the product P from constituent amino acids AA and where it is also subject to degradation.

In Model Capture: Using a Visual Editor
Using the SBW-enabled JDesigner biochemical network editor, a user can create the model using a graphical interface.

Model Visualization and Analysis
Setting the run parameters for the simulation and selecting the output variables in the simulation control GUI allows the user to plot the values of quantities over time. The plot at the right shows how the concentrations of AA and P in the model oscillate over time.

A user can also perform other analyses on the model via SBW, e.g., by invoking the bifurcation analysis module.

SBW in Action: A Sample Session
Here is an example of using several SBW-enabled tools to create and simulate a two-compartment model of a hypothetical single-gene oscillatory circuit.

In this highly simplified model, there is a gene G, which encodes its own repressor and is transcriptionally activated at a constant rate, V. Transcriptional activation of a gene G (which normally involves enzymatic reactions) is summarized here as the production of active RNAP from source material, and degradation to inactive. Transcribed mRNA is then transported from the nucleus into the cytoplasm, where it is translated into the product P from constituent amino acids AA and where it is also subject to degradation.

In Model Capture: Using a Visual Editor
Using the SBW-enabled JDesigner biochemical network editor, a user can create the model using a graphical interface.

Model Visualization and Analysis
Setting the run parameters for the simulation and selecting the output variables in the simulation control GUI allows the user to plot the values of quantities over time. The plot at the right shows how the concentrations of AA and P in the model oscillate over time.

A user can also perform other analyses on the model via SBW, e.g., by invoking the bifurcation analysis module.

Third-Party Modules Available For SBW
- Jarnac, a biochemical simulation package for Windows
- Desmodel, a visual (bio)chemical network layout tool
- Posaconazole, a simple interactive ODE solver
- A stochastic simulator based on Gillespie’s algorithm
- A bifurcation analysis module
- An optimization module
- An SBML validator for checking SBML model EEs
- An Inspector that lists running modules’ & their services

Comming Attractions
More open-source developers are joining the SBW project, and together we are enhancing and extending SBW in many ways. Here is a preview of coming attractions:
- Support for JDK 1.4
- Support for MacOS X
- Bidirectional SOAP-SBW gateway
- JDesigner, a visual (bio)chemical network editor
- New modules, including:
  - Improved generic GUI for simulators
  - Improved, full-featured plot module
  - New simulation engines

How to Get Started With SBW
The SBW version 1.0 package and extensive documentation are available from the project web site, http://www.sbwb.org/. SBW is distributed under the terms of the GNU LGPL.

Acknowledgments
The SBW project is funded by a generous grant from the Japan Science and Technology Corporation.