Model Reproducibility and Part Repositories

Model Sharing Group Presentation

Interagency Modeling and Analysis Group

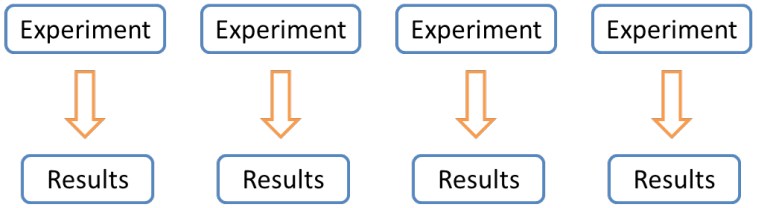
Herbert Sauro
Maxwell Neal
University of Washington

Definitions

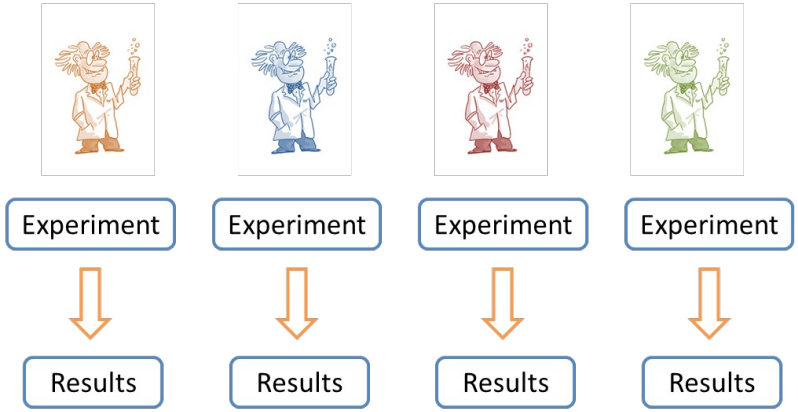
- Repeatability
  - The ability for an individual to show that an experiment, repeated using the same material and equipment, yields the same statistical result.
- Reproducibility
  - The ability for different individuals to show that an experiment repeated using different but similar material and different equipment yields the same statistical result.

Graphically….

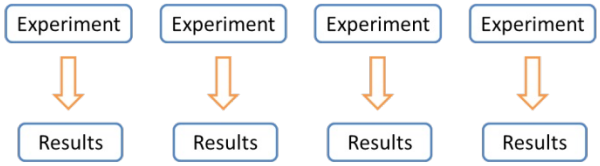
Repeatability



Reproducibility



Model Repeatability

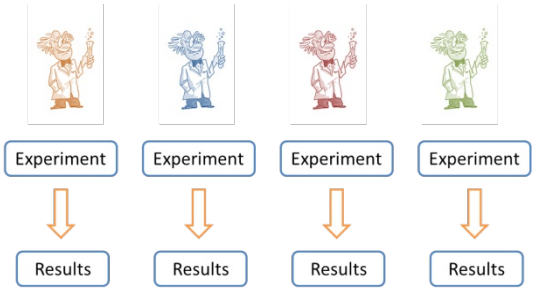


What do we mean by model repeatability?

- Running the model multiple times on the same computer using the same software yields the same result.

- For a stochastic simulation, multiple runs on the same computer will yield the same statistical distribution.

Model Reproducibility



What do we mean by model reproducibility?

- The ability to recreate a published simulation without necessarily using the same software or computer that was used in the original publication.

- At the present time it is a non-trivial exercise to show reproducibility in a published model.

Why Reproducibility is Hard

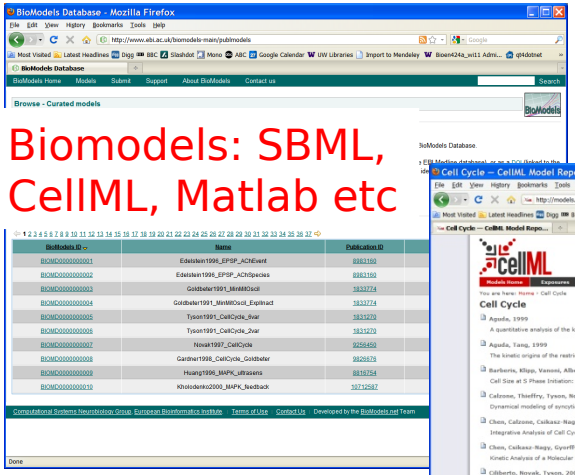
There are a number of reasons why reproducibility of computational models is difficult:

1. The model itself is difficult to extract from a published article.
2. Details of the algorithm and settings that were used are often missing.

We already have a solution to problem 1:

Model repositories.

Model Repositories: Biomodels, CellML and JSim Repositories

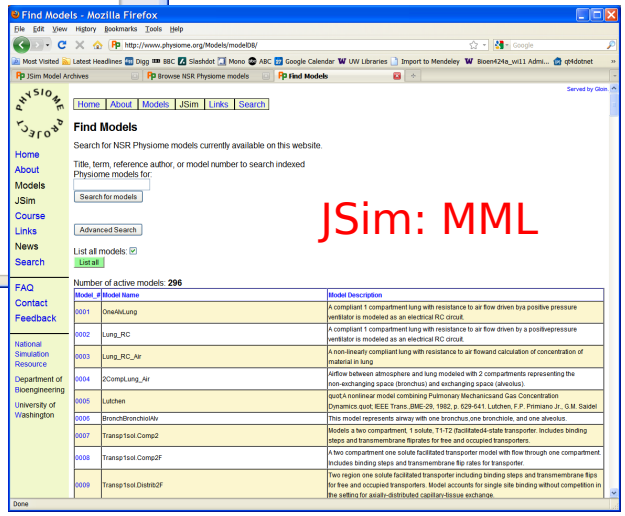


Biomodels: SBML, CellML, Matlab etc

http://www.ebi.ac.uk/biomodels-main/

CellML

http://models.cellml.org/cellml

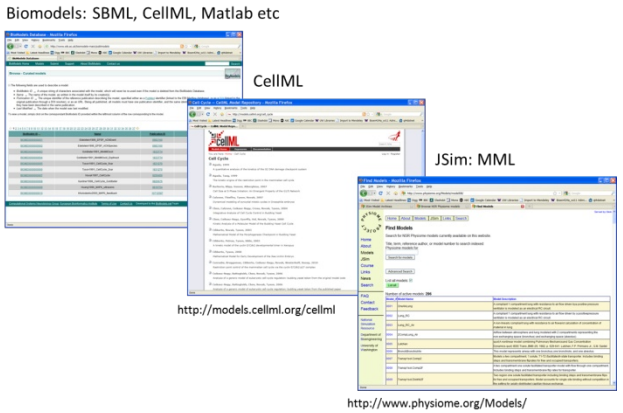


JSim: MML

http://www.physiome.org/Models/

Model Repositories: Biomodels, CellML and JSim Repositories

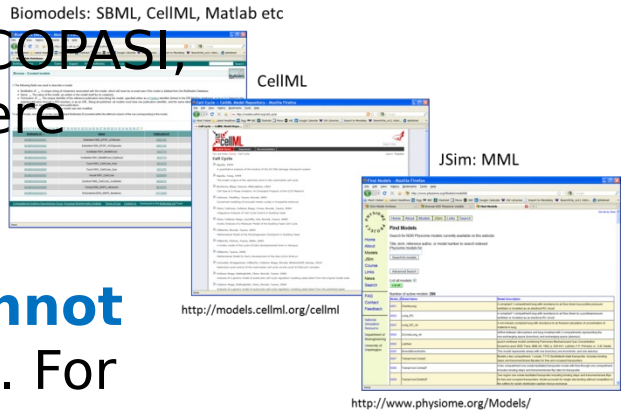
SBML and CellML are **standard**
ts for specifying a model.



do not specify how to generate a simulation (experi

Model Repositories: Biomodels, CellML and JSim Repositories

There are other tools such Jarnac, JDesigner COPASI, TinkerCell, JSim, VCell, CompuCell3D etc where simulations can be specified.



The details of the simulation experiments **cannot** however be easily transferred to another tool. For example a **simulation** experiment in Jarnac cannot be run in COPASI.

All these tools can however generate SBML so that the **models are portable.**

Current Portfolio of Community Standards in Computational Systems Biology

	Model descriptions	Simulations and analysis	Numerical results
Minimal requirements	MIRIAM	MIASE	
Data-models	SBML, BioPAX, CellML, SBGN	SED ML	NuML? **SBRML**
Terminologies	SBO Rate Kinetics	KISAO Algorithms	TEDDY Behaviors

odified from Nicolas Le Novère **SBRML = Systems Bio Results (data) Markup Lang**

SBRML: **Systems Biology Results Markup Language**

Can be use to store information such as:

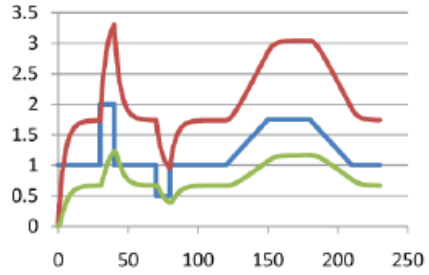
1. Time Course Simulations
2. Steady State Results
3. Enzyme Kinetic Data
4. Experimental Data
5. Results of parameter scans
6. Who generated the data?
7. etc. – very flexible proposal

eveloped by Mendes Manchester Group in the UK

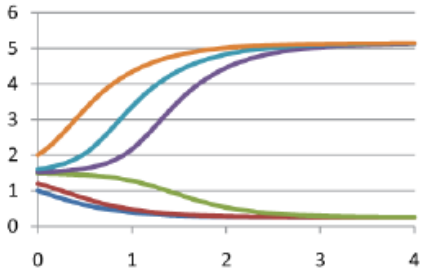
SED-ML: **Simulation Experiment Description Language**

SED-ML: **Simulation Experiment Description Language**

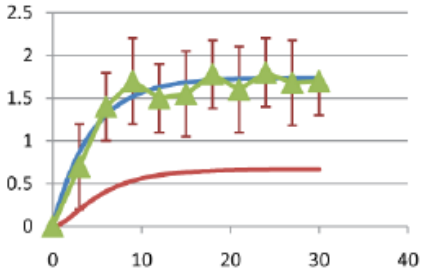
Complex Series of Pulses and Ramps



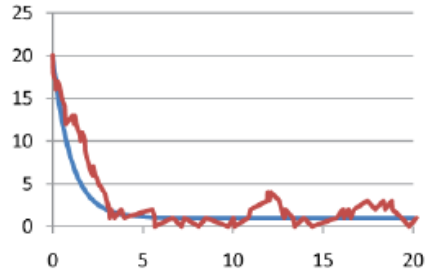
Multiple Initial Conditions



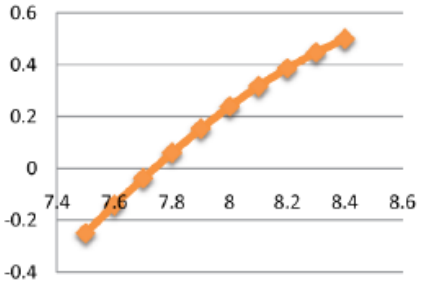
Optimization of Simulation to Experimental Data



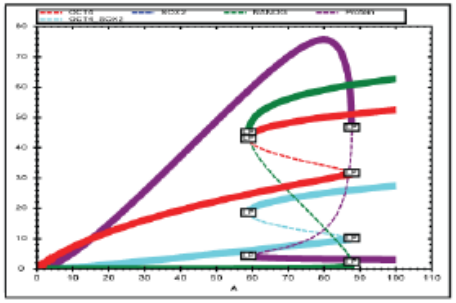
Comparing Deterministic and Stochastic Models



Eigenvalues Versus Parameter



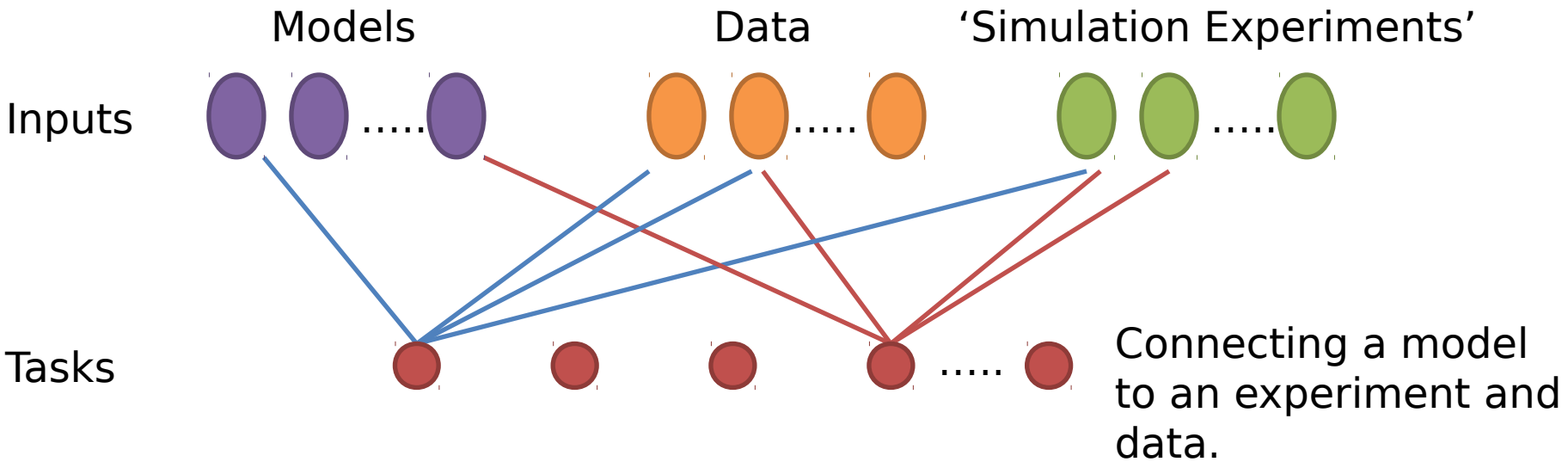
Bifurcation Analysis



Selection of use case simulations that apply to SEDML

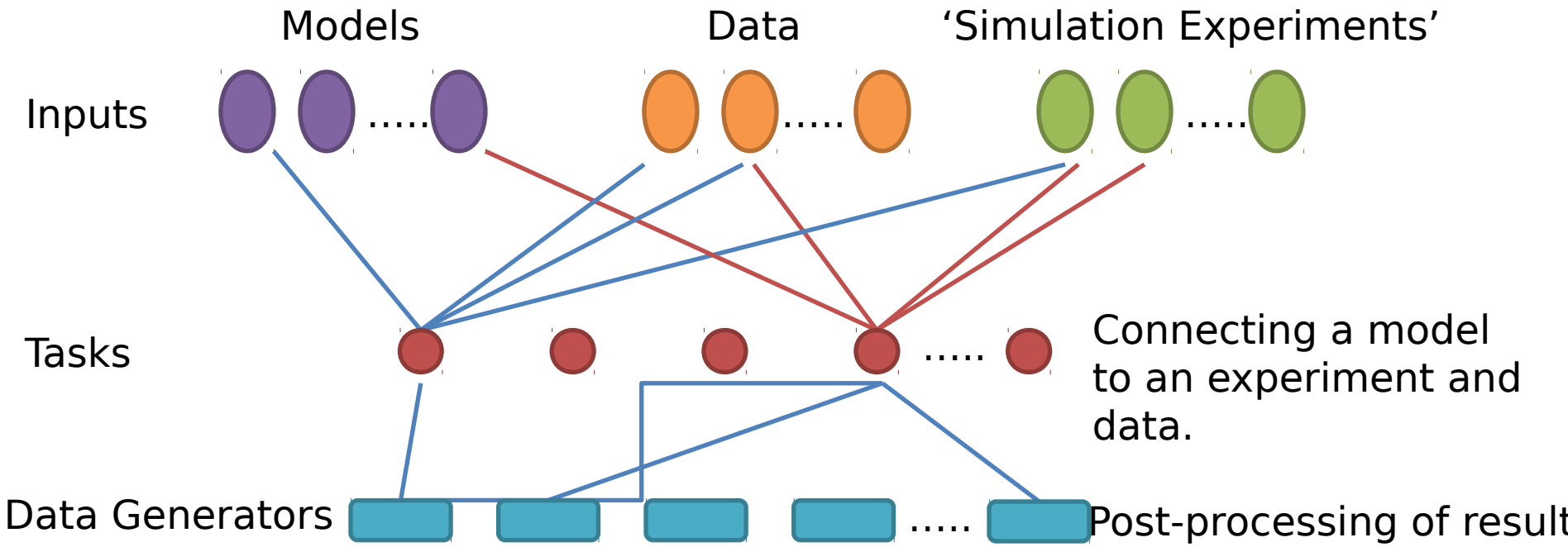
SED-ML: **Simulation Experiment Description Language**

SED-ML: **Simulation Experiment Description Language**

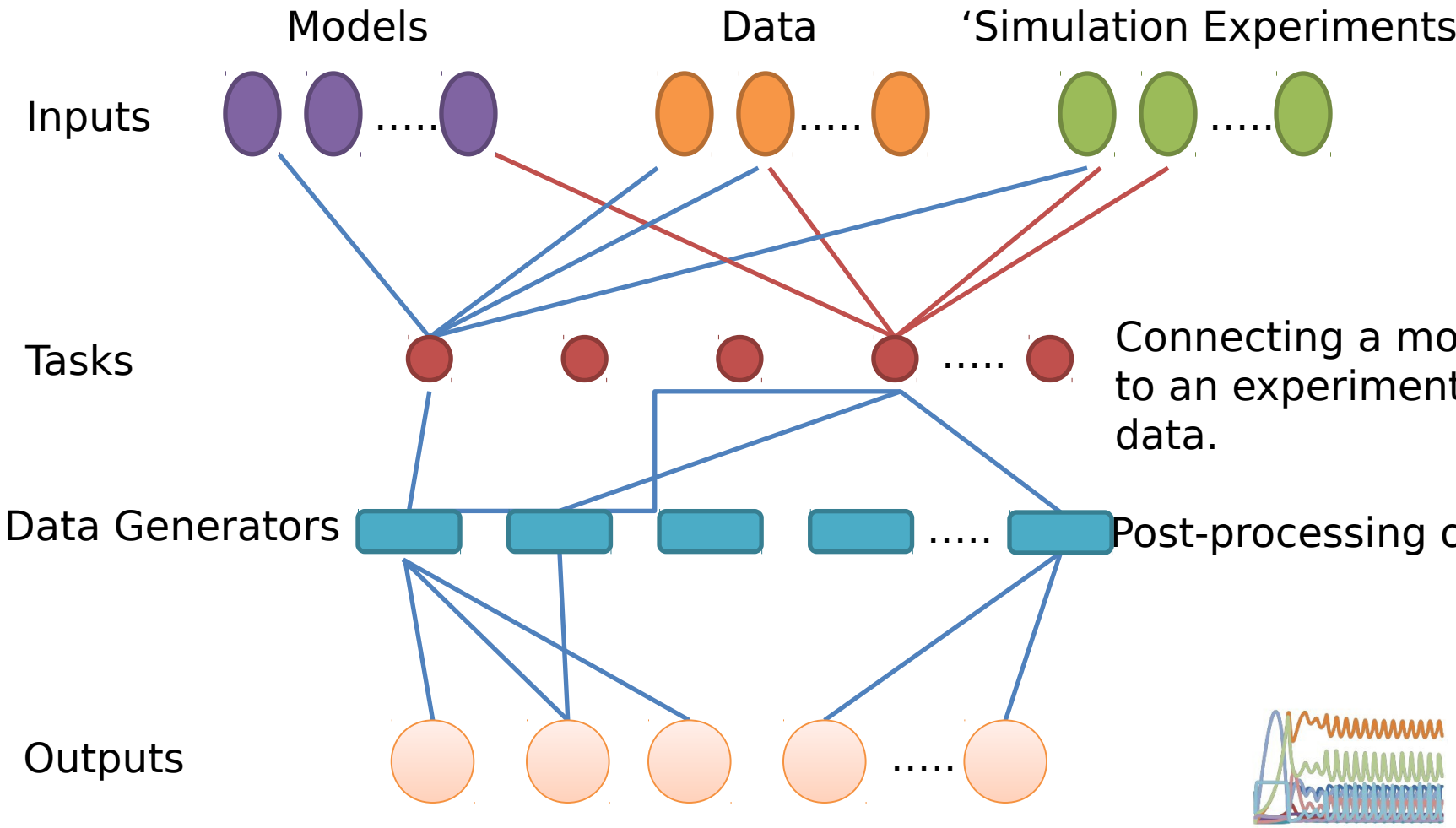


Note: One model per task but multiple data and experiments per task.

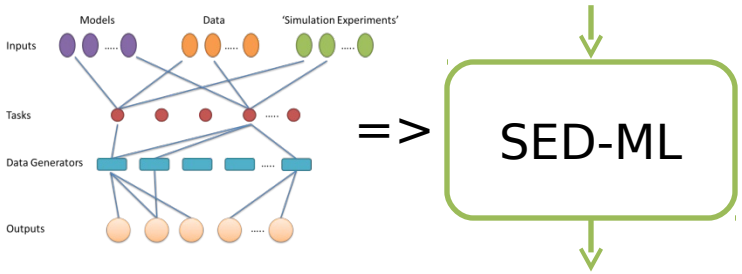
SED-ML: **Simulation Experiment Description Language**

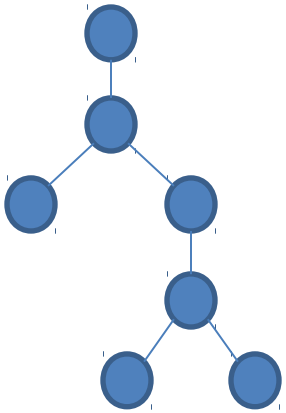


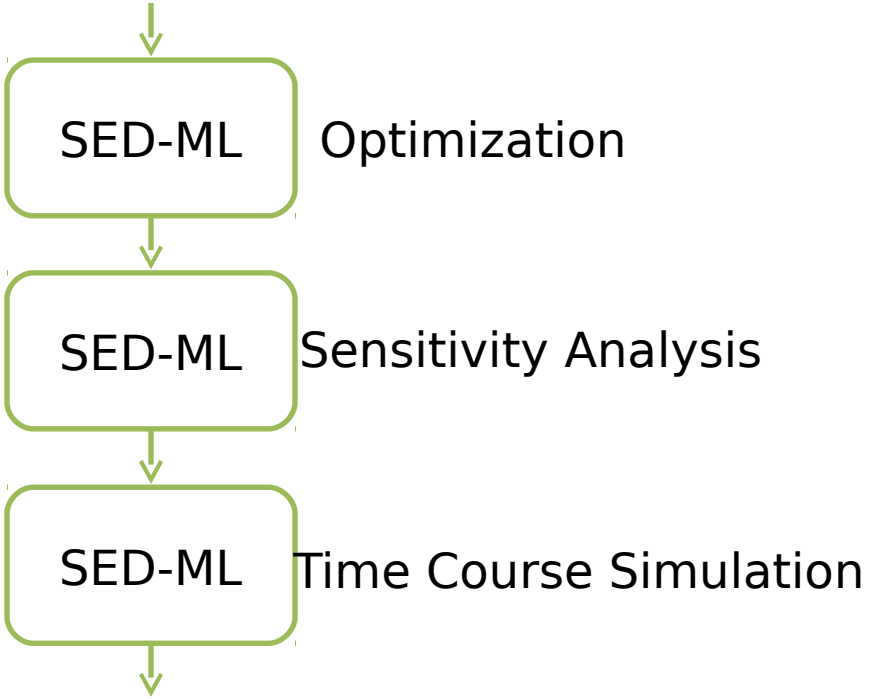
SED-ML: **Simulation Experiment Description Language**



Concatenating Experiments



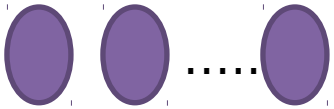




SED-ML: **Simulation Experiment Description Language**

Inputs

Models

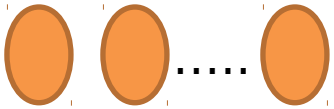


Models:

SBML
CellML
XMML

Currently must be expressible in **XML**,
i.e cannot be Matlab,
Java etc. Must be able to annotate and must have a

Data

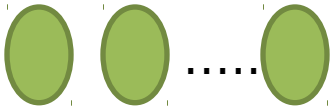


Data:

No standard format yet but **SBRML** is a possibility.

Could be virtual patients, time course data for fitting, flux balance data, Etc.

'Simulation Experiments'



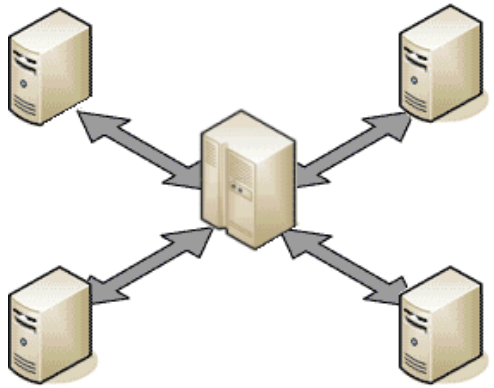
Experiments:

Perturbations,
Time Course,
Steady State,
Numerical Methods Settings,
Parameter Scans,
Optimization,
Sensitivity Analysis,
Bifurcation Analysis,
etc

SED-ML: **Simulation Experiment Description Language**

Because the models, data and experiments are based on XML, the different files need not reside on the same computer. For example it would be possible to reference a model remotely from Biomodels or the Cellml repository. The same goes for data and experiments.

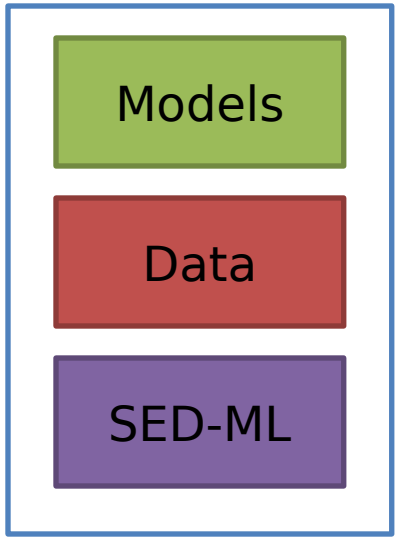
In the case of large datasets it might not be practical to have the data on a local machine and remote access it more convenient.



http://techiesniffer.blogspot.com/2011/03/client-server-

SED-ML: **Archival Format**

Rather than have separate files for models, data and experiments, it is also proposed to have a single archival file. This file will be a zip file containing models (SBML, etc), Data (SBRML) and Experiments (in SED-ML).



Zip File

Ancillary Efforts

KiSAO: Kinetic Simulation Algorithm Ontology

KiSAO can be used to identify both the algorithm used and the initial setup. For example what ODE solver was used and what tolerances etc where specified.

What does SED-ML Look like?

```
<?xml version="1.0" encoding="utf -8"?>
<sedML xmlns="http://sed-ml.org/" xmlns:math="http://www.w3.org/1998/Math/MathML" level="1" version="1">
<notes>
    <p xmlns="http://www.w3.org/1999/xhtml">Comparing Limit Cycles and strange attractors for oscillation in Drosophila</p>
</notes>
<listOfSimulations>
    <uniformTimeCourse id="simulation1" initialTime="0" outputStartTime="0" outputEndTime="380" numberOfPoints="1000">
        <algorithm kisaoID="KISAO:0000019"/>
    </uniformTimeCourse>
</listOfSimulations>
<listOfModels>
    <model id="model1" name="Circadian Oscillations" language="urn:sedml:language:cellml" source="http://
models.cellml.org/workspace/leloup_gonze_goldbeter_1999/@@rawfile/7606
a47e222bc3b3d9117baa08d2e7246d67eedd/leloup_gonze_goldbeter_1999_a.cellml"/>
    <model id="model2" name="Circadian Chaos" language="urn:sedml:language:cellml" source="model1">
        <listOfChanges>
            <changeAttribute target="/cellml:model/cellml:component[@name='MT']/cellml:variable[@name='vmT ']/
  @initial_value" newValue="0.28"/>
            <changeAttribute target="/cellml:model/cellml:component[@name='T2']/cellml:variable[@name='vdT ']/
  @initial_value" newValue="4.8"/>
        </listOfChanges>
    </model>
</listOfModels>
<listOfTasks>
<task id="task1" name="Limit Cycle" modelReference="model1" simulationReference="simulation1"/>
<task id="task2" name="Strange attractors" modelReference="model2" simulationReference="simulation1"/>
</listOfTasks>
<listOfDataGenerators>
Etc....
```

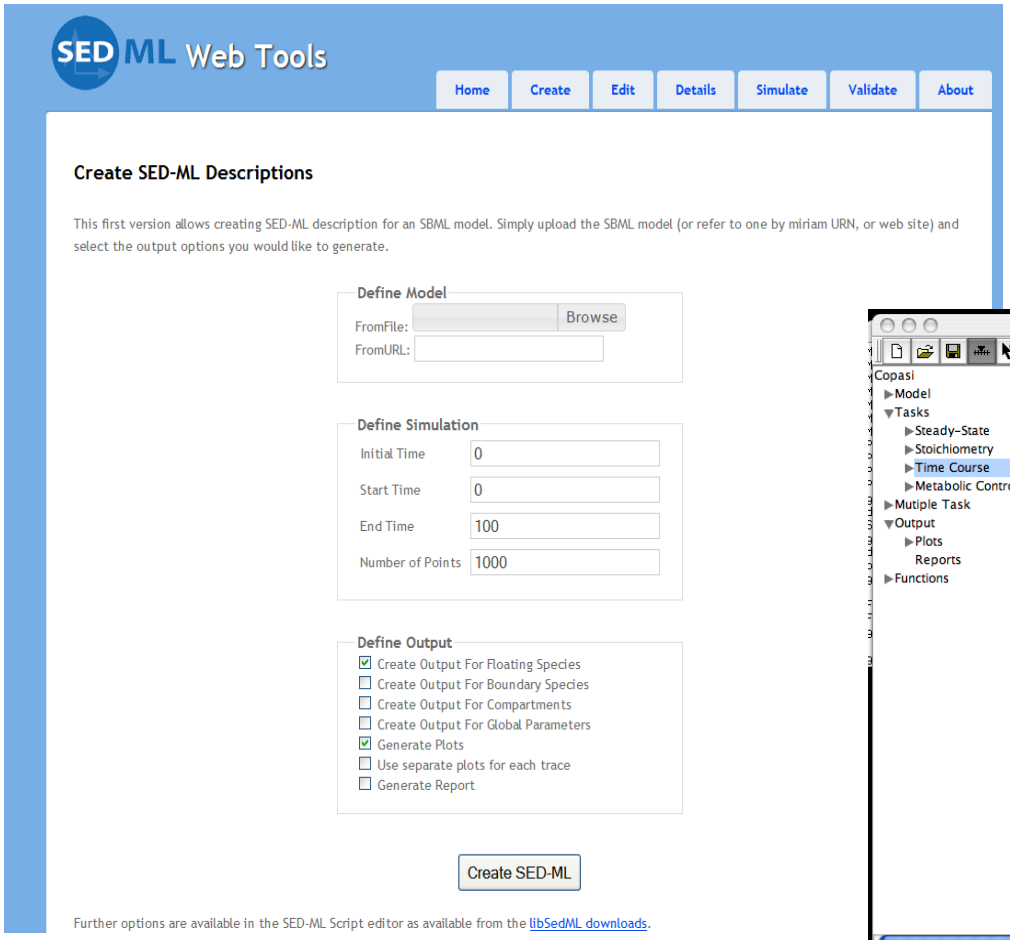
What about the User?

Obviously the user isn’t going to write XML

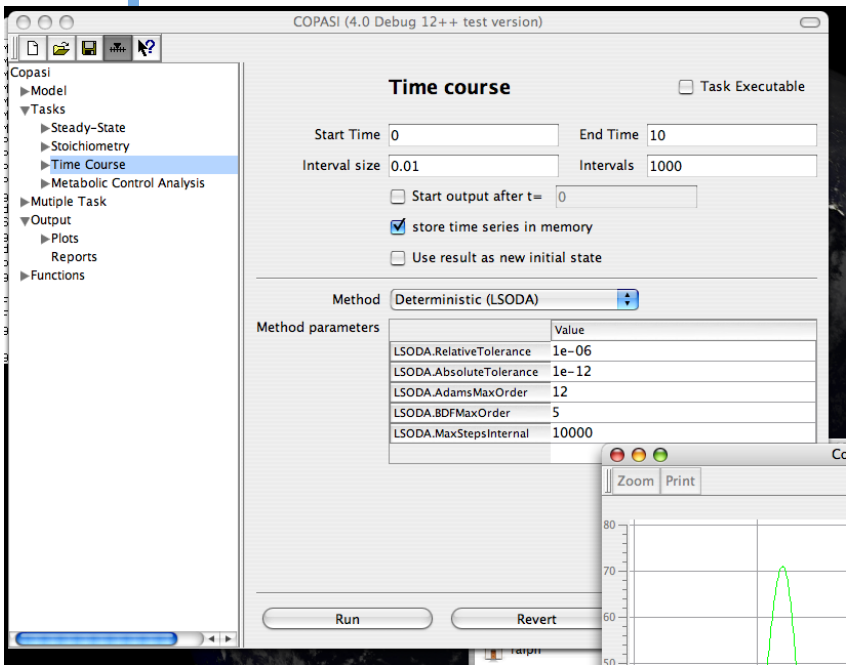
What might the user experience be?

There are at least three approaches here

Forms Based Entry of SED-ML



http://sysbioapps.dyndns.org/SED-ML Web Tools/
Frank Bergmann

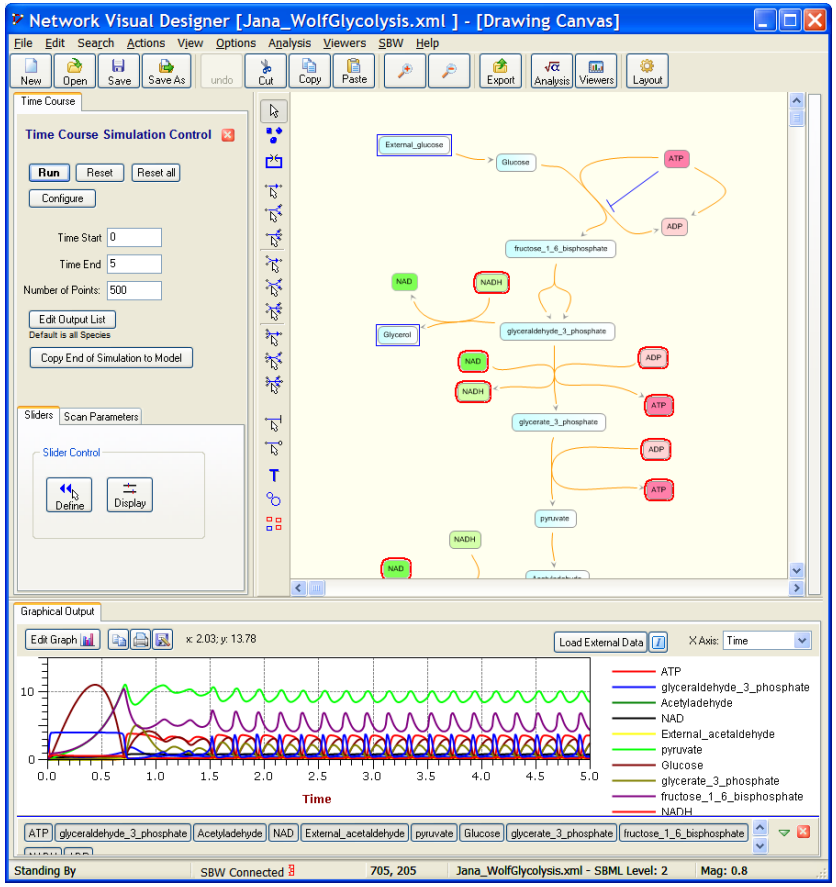


COPASI (copasi.org)

Script Based Definition of SED-ML

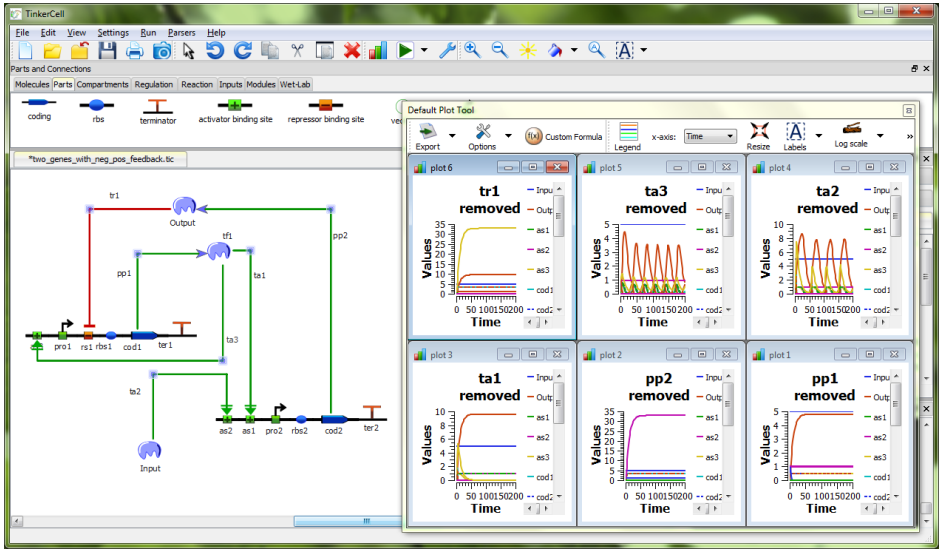
```
model myModel (Xo, X1)

  var S1, S2;   ext Xo, X1;   // Declare variables and boundary species

  Initial                          // Set up initial conditions and parameter values
    Xo = 10.0; X1 = 0.0;
    k1 = 0.1;  k2 = 0.3;
    start = 5; duration = 2; slope = 0.2

  Signals
     Xo = ramp (start, duration, slope);

  Events
     when S1 > 5 do
        k1 = k1 / 2;

  Network                          // Define the biochemical network
     Xo -> S1; k1*Xo;
     S1 -> S2; k2*S1;
     S2 -> X1; k3*S2;
end;

// Specify Simulation Experiment
m1 = runSimulation (0, 10, 100);
reset;
k1 = k1 * 2;
m2 = runSimulation (0. 10. 100);
output (m1, m2);
```

Graphical UI Tracking



JDesigner (sys-bio.org)

All operations are tracked by the software so that model changes and simulations can be replayed.



Tinkercell (tinkercell.com)

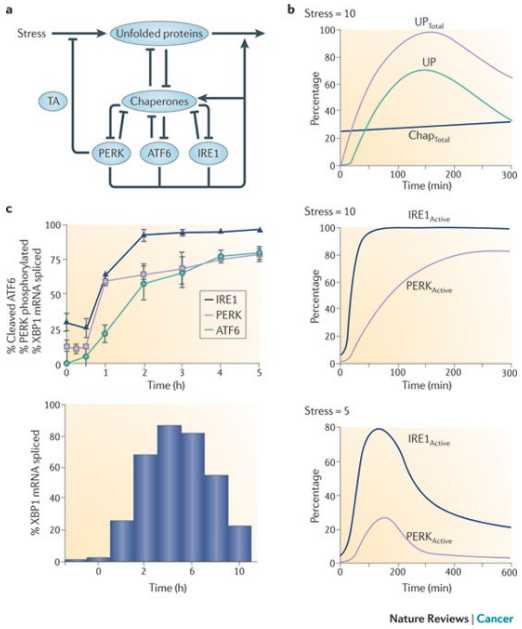
What’s in it for me?

Reproducibility!

What’s in it for me?

Download a pdf article from pubmed

ed protein response (UPR), y axis % unfolded protein, total chaperone and free



From the figure extract all the information required to reproduce the three simulation experiments at the three stress levels:

1. Model including the SBGN network diagram
2. Data (possibly for fitting or comparison)
3. SED-ML that describes three experiments.

Load into your favorite tool to recreate figure.

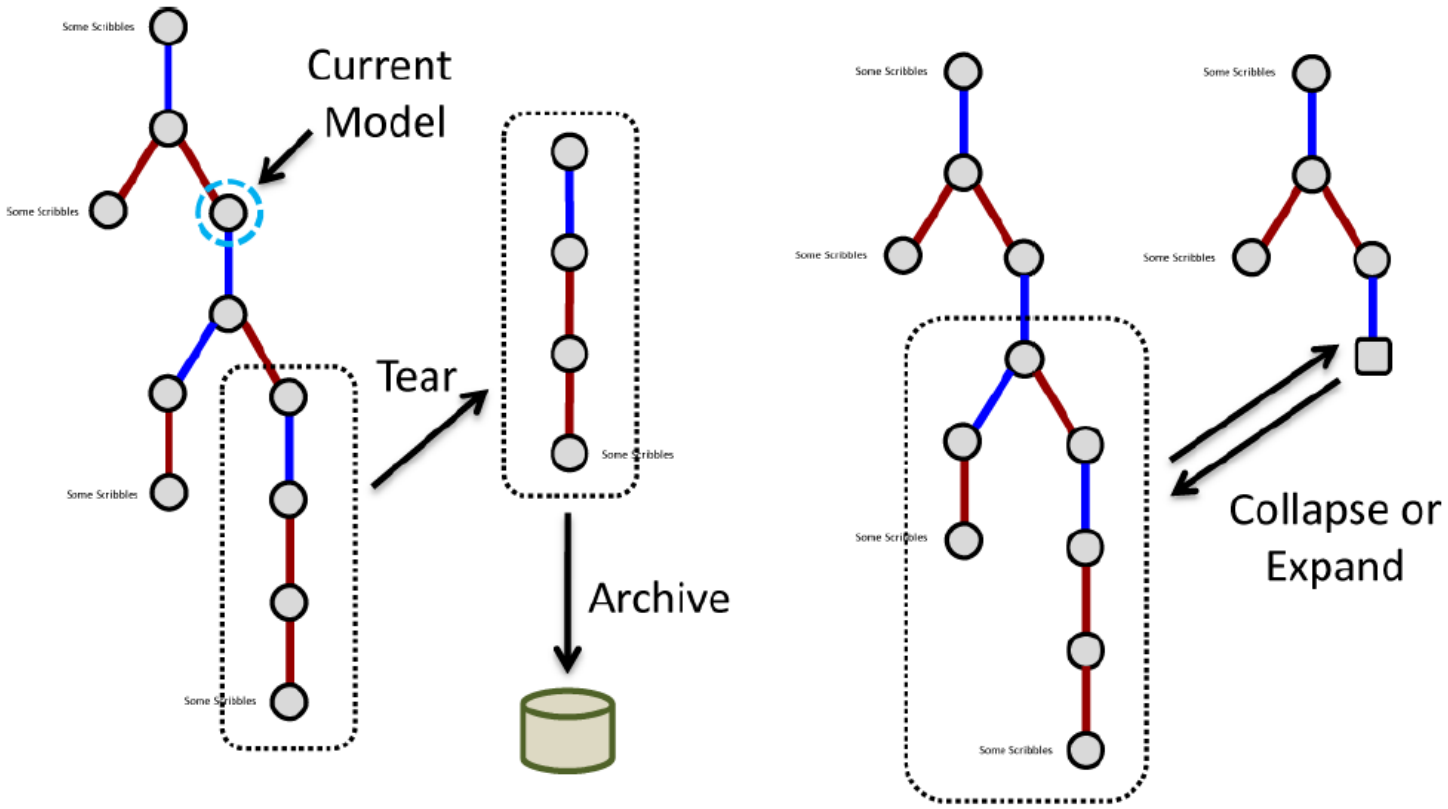
Dynamic modelling of oestrogen signalling and cell fate in breast cancer cells, Tyson el al, Nature Rev Cancer, 11, 523-532 (2011)

But there’s more...

Once we have the ability to encode simulation experiments there are a few other related things we can think about doing, these include:

1. Tracking Model Changes
2. Recording Simulations to Exchange or Replay
3. Versioning
4. Unit Tests
5. Multiple data sets, eg support virtual patients
6. Supporting Animations
7. Formalize and automate simulator testing

Versioning and Tracking



SED-ML

- SEDML homepage:
  http://www.biomodels.net/sedml

- SEDML at Sourceforge:
  https://sourceforge.net/projects/sedml
  Support library to read and write SEDML, developed by VCell and CSBE (Edinburgh) groups - jlibSEDML

- SEDML mailing list:

  sedmldiscuss@lists.sourceforge.net

Prototypes developed by:

Frank Bergmann (part of SBW Project)
Jacky Snoep (JWS Online Simulator)
Richard Adams (SBSI: Edinburgh)
Ion Moraru (VCell) via SBW grant
Peter Hunter (PCEnv)

Future developments

Jim Bassingthwaighte (JSim) via SBW grant.
Sven Sahle (COPASI, EU)
Nicolas Le Novere (Biomodels, EU/UK)

Community Effort

US:

- Herbert Sauro (UW, SBW)
- Ion Moraru (Connecticut, VCell)
- Jim Bassingthwaighte (UW, JSim)
- Frank Bergmann (Caltech, SBW)
- Mike Hucka (Caltech, SBML)

Europe:

- Dagmar Waltemath (Rostock, SEDML)
- Nicolas Le Novere (EBI, Biomodels)
- Richard Adams (Edinburgh, SBSI)
- Sven Sahle (Heidelberg, COPASI)
- Henning Schmidt (SB Toolbox)
- Fedor Kolpakov (BioUML)

New Zealand

- Andrew Miller (Auckland, CellML)
- David Nickerson (Auckland, CellML)

Funding:

wellcome trust

EPSRC
Engineering and Physical Sciences Research Council

EMBL-EBI

BBSRC
bioscience for the future

NIGMS

Component Repositories

Possibly one of the hardest things about modeling a cellular network is researching the rate laws and parameters that one should use to build the model.

What if there were a repository of ready made parts that could be dropped into a model?

Exploiting Biomodels

The biomodels repository has almost 400 curated models, many of which are annotated.

We can take the annotations and use them to break up every biomodel into its constituent parts.

That will result in over 4000 parts that could be used in new models.

What does a biomodels part look like?

Reaction: phosphofructokinase ← Name of the enzyme

Reactants:
hexose monophosphate (HMP)
ATP (ATP)

Products:
fructose 1,6-bisphosphate (Fru16P2)

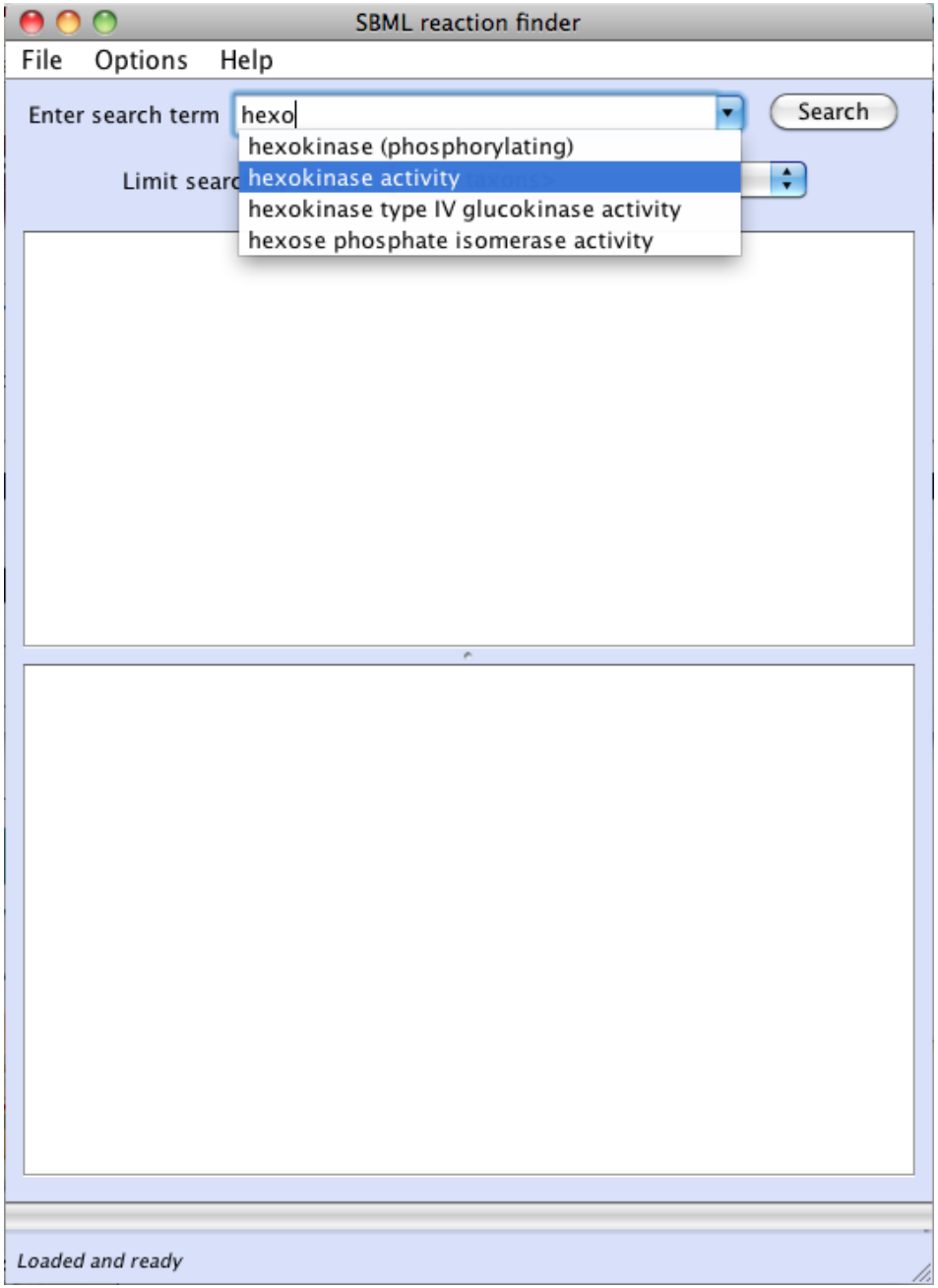
Modifiers:

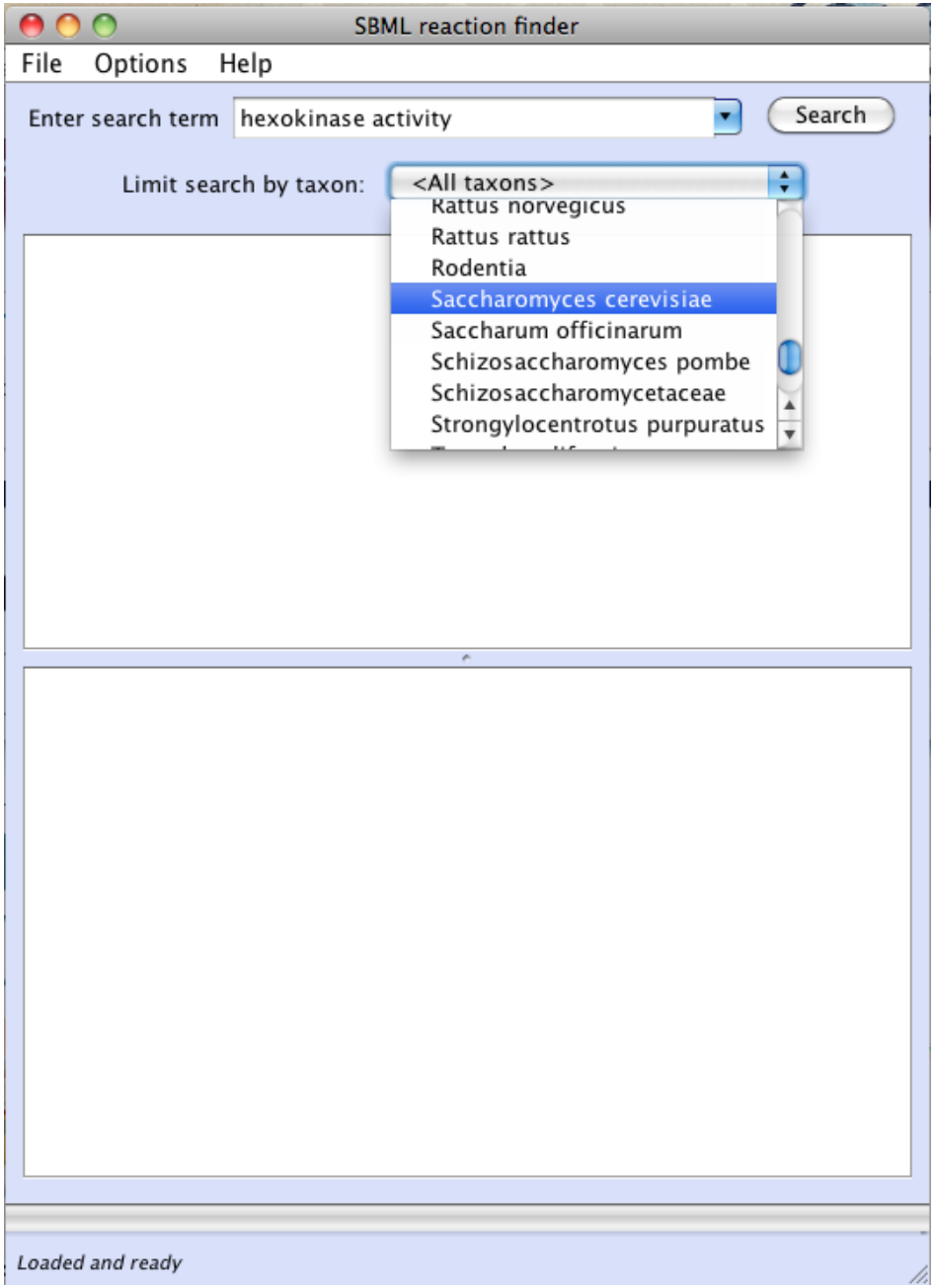
Kinetic Law:
cell * VPFK * gR * lambda1 * lambda2 * R / (pow(R, 2) + L * pow(T, 2)) ← The rate law

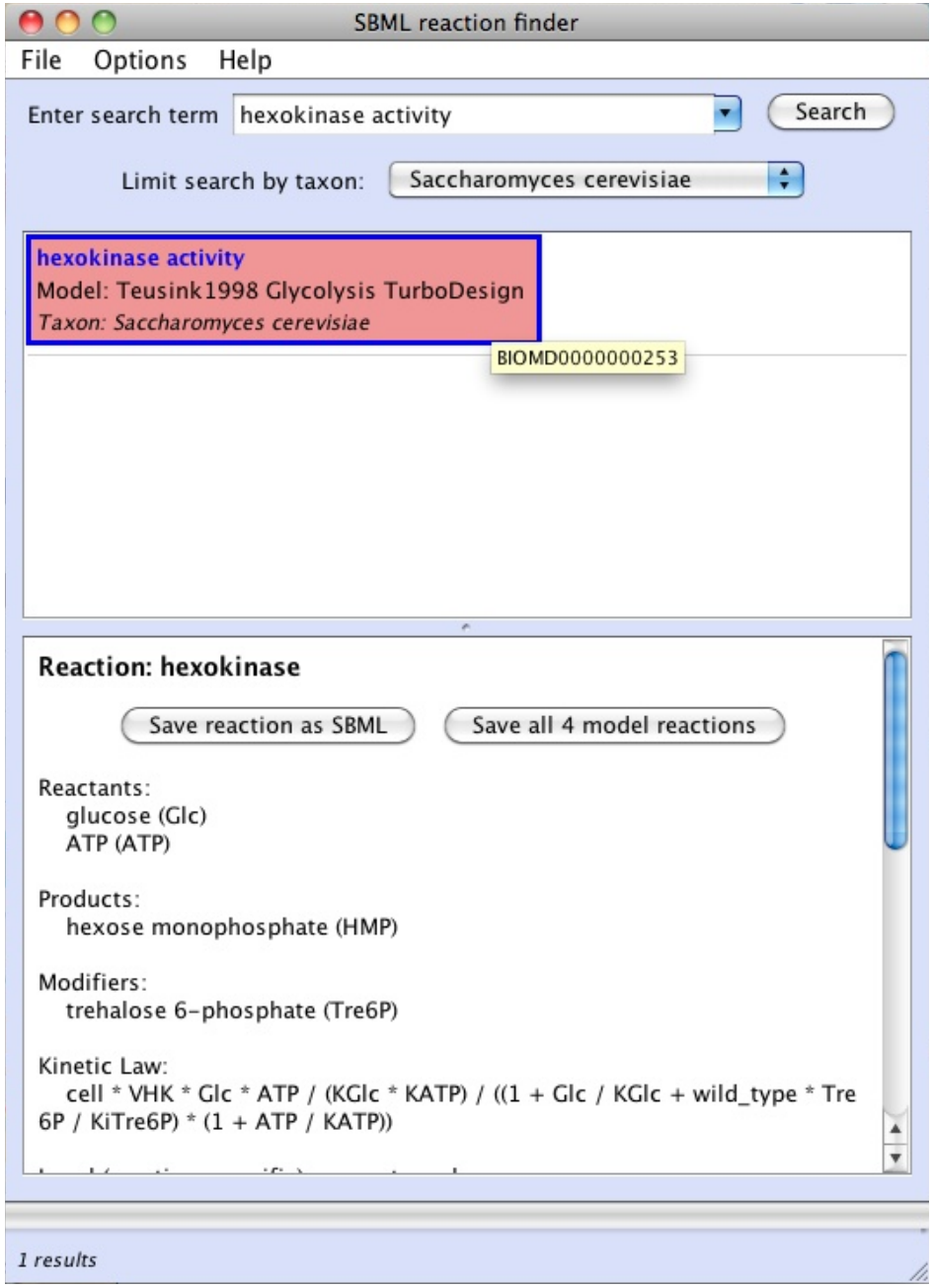
Local (reaction-specific) parameter values:
VPFK = 30.0 mM_per_min ← The parameter values that were used
etc

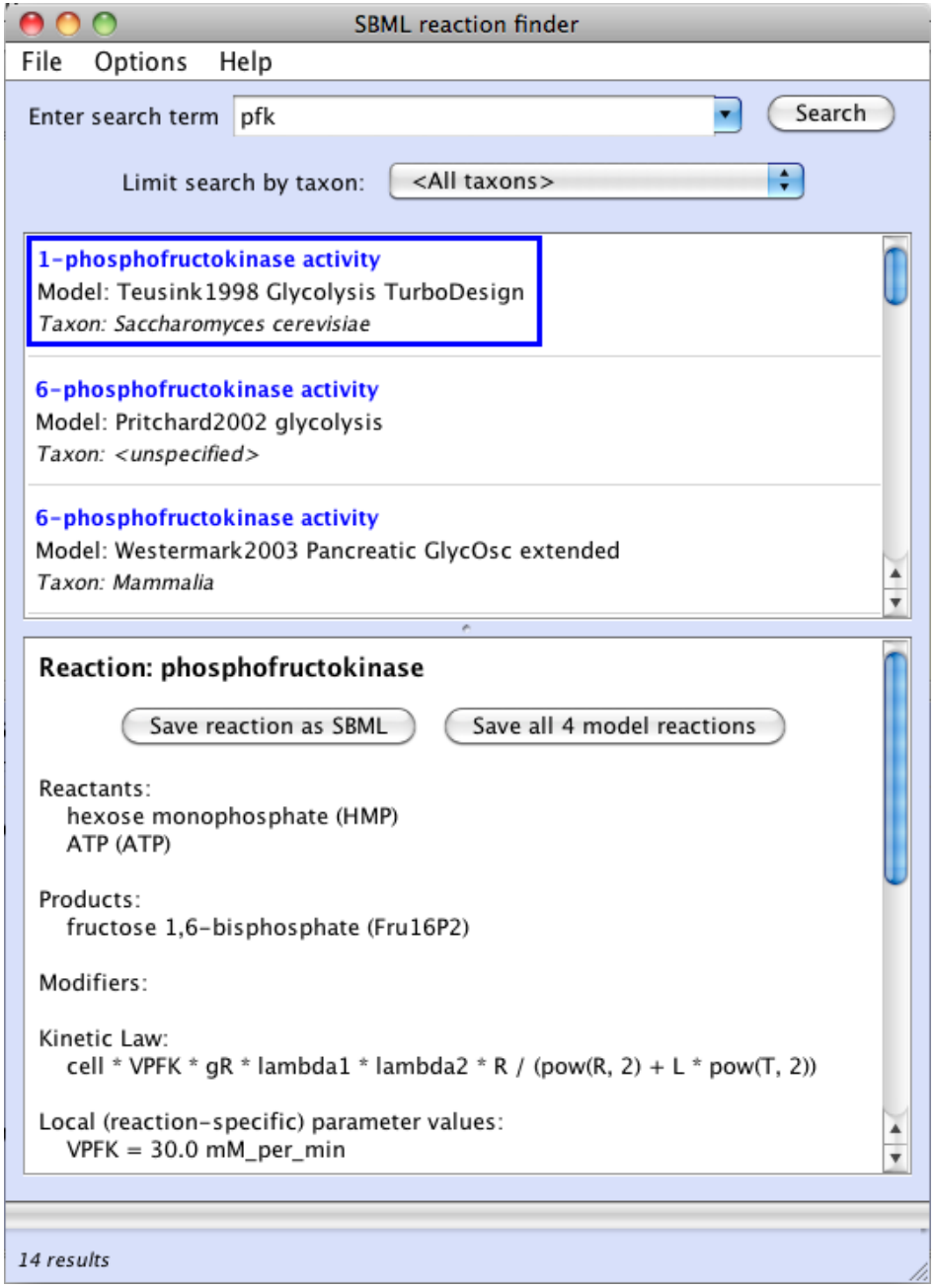
How was it done

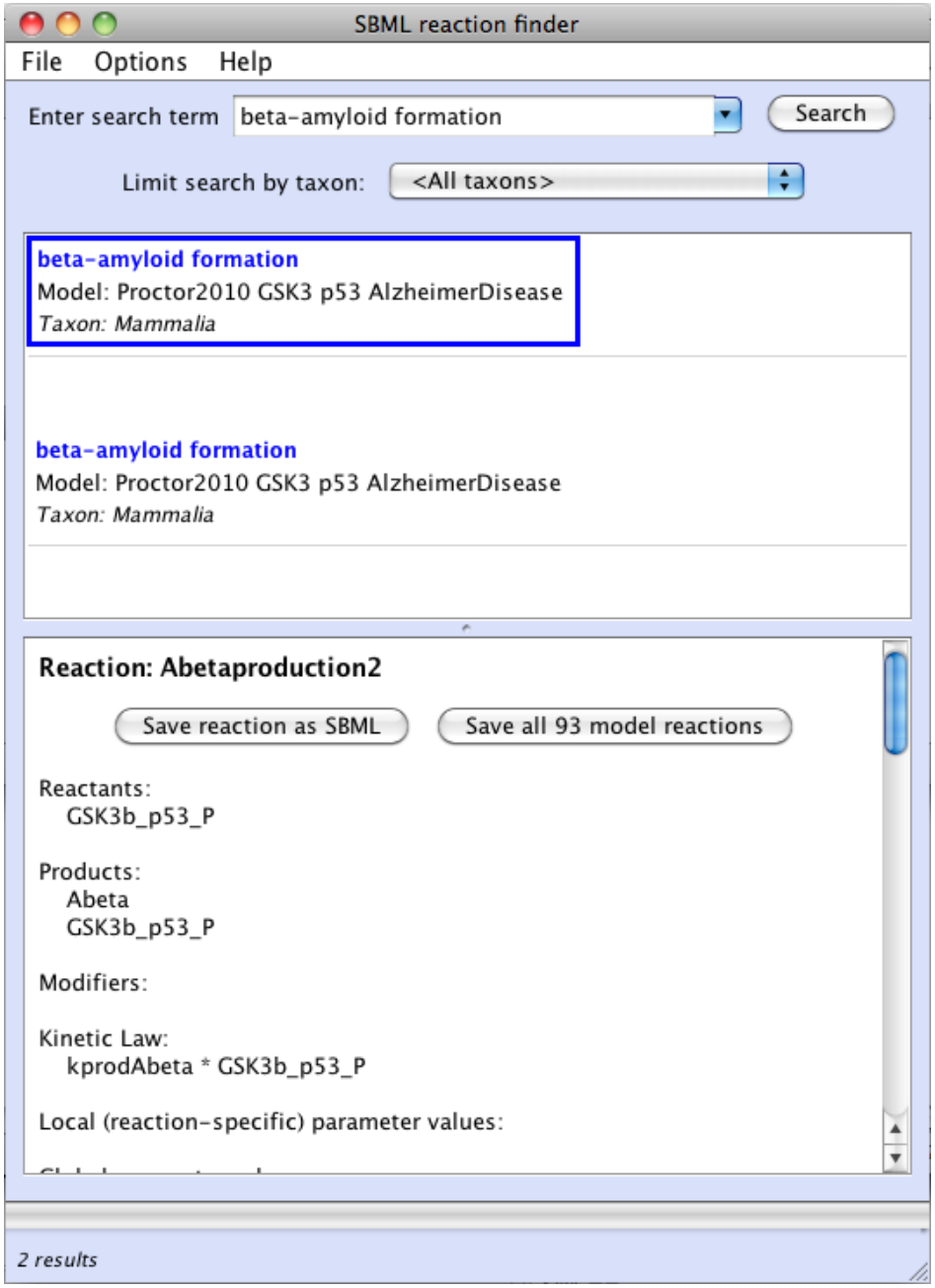
- A knowledge base was created that represents each individual reaction that is annotated against a GO term in the BioModels repository.
- Each individual reaction is associated with it's GO term, its synonyms, its codename in the SBML model, and the SBML model free-text description.
- A search using the tool sends a pattern match query across these attributes of the reactions.
- The tool accesses the repository itself so that it is always up to date.
- This tool can be incorporated in to simulators allowing users to browse for suitable parts they can include in their model.
- We’d like to do the same for CellML and the JSim repositories so that users can browse for parts related to physiological systems.


SBML reaction finder
File
Options
Help
Enter search term
hexo
Search
hexokinase (phosphorylating)
hexokinase activity
hexokinase type IV glucokinase activity
hexose phosphate isomerase activity
Limit searc
Loaded and ready


SBML reaction finder
File
Options
Help
Enter search term
hexokinase activity
Search
Limit search by taxon:
<All taxons>
Rattus norvegicus
Rattus rattus
Rodentia
Saccharomyces cerevisiae
Saccharum officinarum
Schizosaccharomyces pombe
Schizosaccharomycetaceae
Strongylocentrotus purpuratus
Loaded and ready


SBML reaction finder
File
Options
Help
Enter search term
hexokinase activity
Search
Limit search by taxon:
Saccharomyces cerevisiae
hexokinase activity
Model: Teusink1998 Glycolysis TurboDesign
Taxon: Saccharomyces cerevisiae
BIOMD0000000253
Reaction: hexokinase
Save reaction as SBML
Save all 4 model reactions
Reactants:
glucose (Glc)
ATP (ATP)
Products:
hexose monophosphate (HMP)
Modifiers:
trehalose 6-phosphate (Tre6P)
Kinetic Law:
cell * VHK * Glc * ATP / (KGlc * KATP) / ((1 + Glc / KGlc + wild_type * Tre 6P / KiTre6P) * (1 + ATP / KATP))
1 results


SBML reaction finder
File
Options
Help
Enter search term
pfk
Search
Limit search by taxon:
<All taxons>
1-phosphofructokinase activity
Model: Teusink1998 Glycolysis TurboDesign
Taxon: Saccharomyces cerevisiae
6-phosphofructokinase activity
Model: Pritchard2002 glycolysis
Taxon: <unspecified>
6-phosphofructokinase activity
Model: Westermark2003 Pancreatic GlycOsc extended
Taxon: Mammalia
Reaction: phosphofructokinase
Save reaction as SBML
Save all 4 model reactions
Reactants:
hexose monophosphate (HMP)
ATP (ATP)
Products:
fructose 1,6-bisphosphate (Fru16P2)
Modifiers:
Kinetic Law:
cell * VPFK * gR * lambda1 * lambda2 * R / (pow(R, 2) + L * pow(T, 2))
Local (reaction-specific) parameter values:
VPFK = 30.0 mM_per_min
14 results


SBML reaction finder
File
Options
Help
Enter search term
beta-amyloid formation
Search
Limit search by taxon:
<All taxons>
beta-amyloid formation
Model: Proctor2010 GSK3 p53 AlzheimerDisease
Taxon: Mammalia
beta-amyloid formation
Model: Proctor2010 GSK3 p53 AlzheimerDisease
Taxon: Mammalia
Reaction: Abetaproduction2
Save reaction as SBML
Save all 93 model reactions
Reactants:
GSK3b_p53_P
Products:
Abeta
GSK3b_p53_P
Modifiers:
Kinetic Law:
kprodAbeta * GSK3b_p53_P
Local (reaction-specific) parameter values:
2 results

How to get hold of SRF

SBML Reaction Finder

Description:

http://sbp.bhi.washington.edu/projects/sbmlrxnfinde

Downloads:

http://sf.net/projects/sbmlrxnfinder/