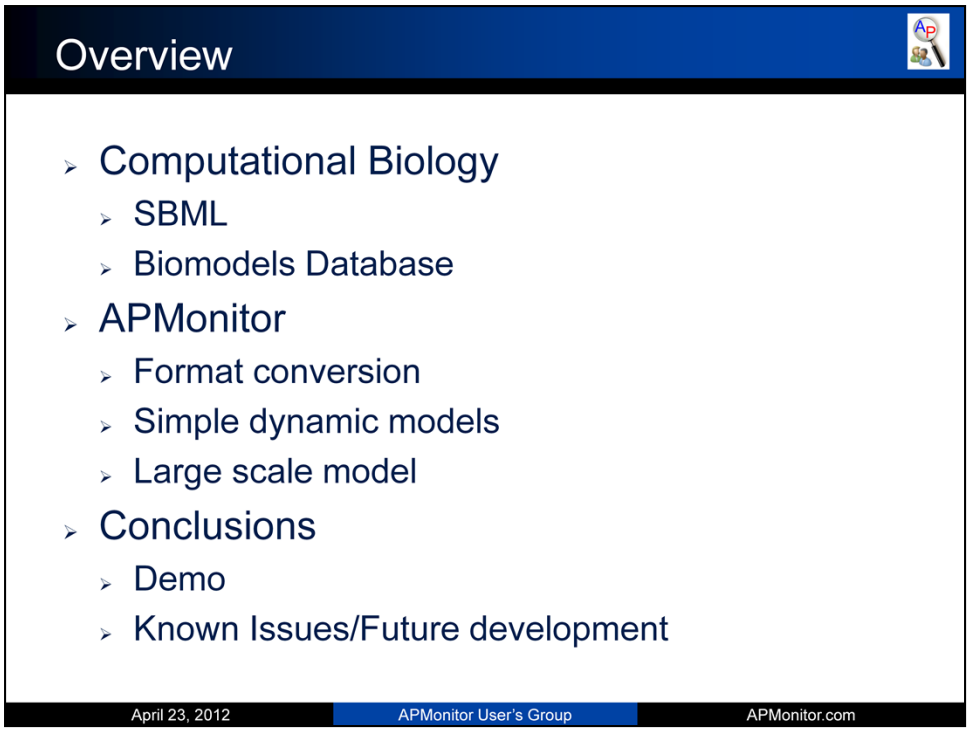
Biological Modeling using APMonitor

David Grigsby
Brigham Young University

23 Apr 2012



For this presentation, I will give a brief discussion on why computational biology is useful, how some biological models are stored and where you can find them.

Then I will discuss the work I have done in creating a format conversion utility in Java to enable APMonitor to be used to simulate and perform parameter estimation on compiled biological models. In my research I have worked with both small and large scale models and some examples will be shown to display APMonitors capabilities and some limitations of my converter tool.

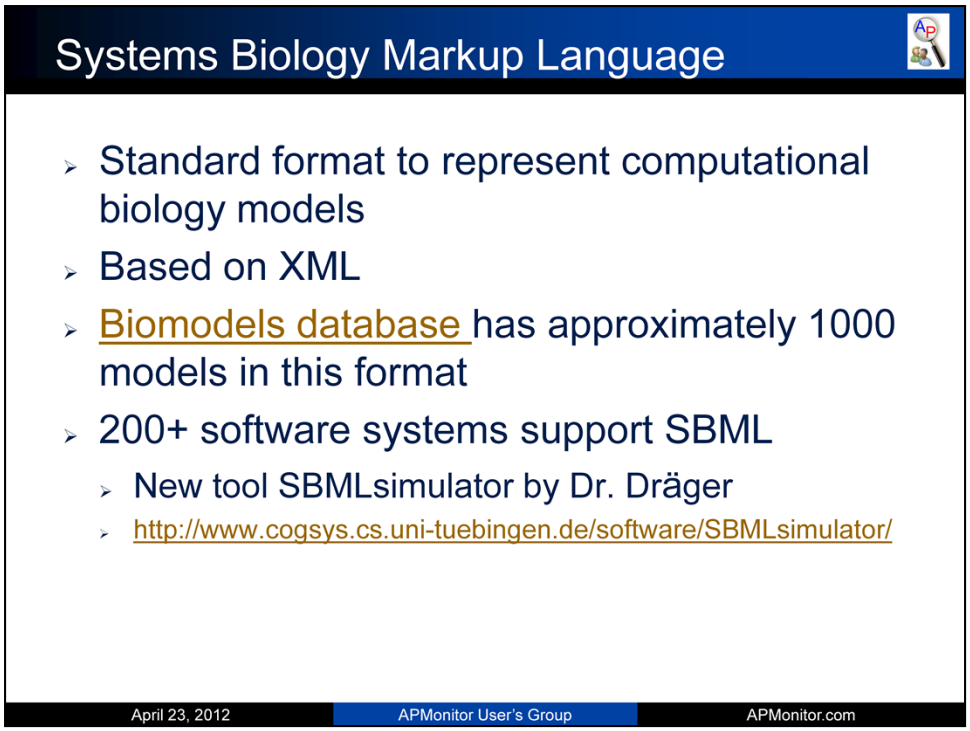
I will conclude with a live demonstration of how to use the conversion utility and the APMonitor software in a web interface as well as explaining how you can become involved with APMonitor. Also, I will explain current issues we have noticed and how we hope to overcome them.

Importance of Computational Biology

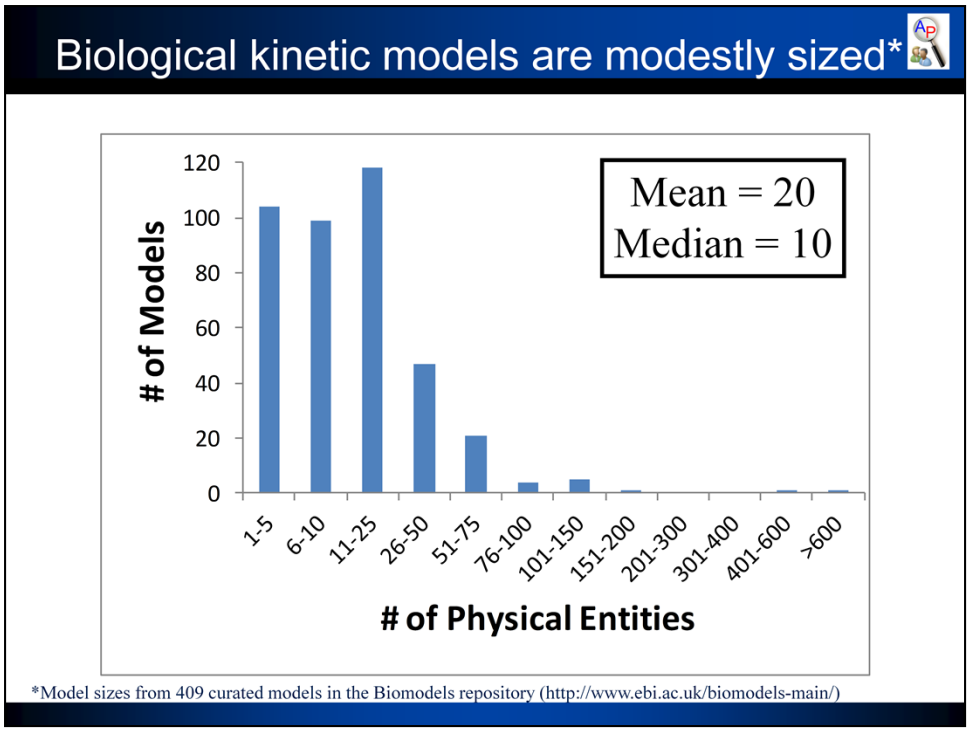
- Growing area to guide drug development and treatment
- Key to advancement of neuroscience and genome modeling

April 23, 2012 APMonitor User's Group APMonitor.com

Computation Biology is a broad field that is basically the application of modeling and computational simulation to biological and social systems. It is a growing area that can help guide drug development to reduce the number of necessary clinical trials and improve effectiveness of drugs. It can also be applied to complex biological systems in order to provide a better understanding of systems such as neuroscience and genome modeling.

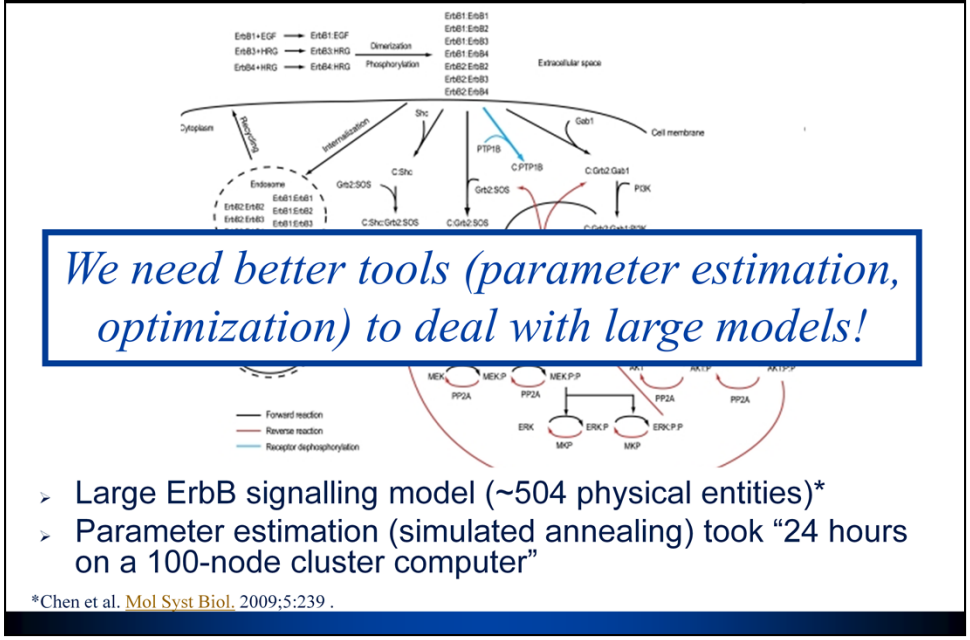


How are many of these computational biology models stored? The SBML: Systems Biology Markup Language. It is a standard format that many software systems support to help biological modelers have a uniform system to publish their models. Currently the biomodels database has over 400 curated models and 400 submitted models in the SBML format. However, even though over 200 software systems support the creation and reading of these models, there are not as many tools, MATLAB, octave, biopax, that can be used to analyze and perform efficient parameter estimation on these models. One software tool of note is the SBMLsimulator developed recently by Dr. Dräger and his team.



A little more about the biomodels database itself….most published models are moderately sized. Since biological systems are very complex it can be very difficult to create very large-scale models with accuracy. I performed a count of all the physical entities in the curated section of the biological models database and found that the average size model contained 20 physical entities, but most were smaller and many submissions were slightly altered replicates for the same publication. However, if biological systems are so complex there is a potential that many more physical entities could be measured and modeled to produce more large scale and potentially more powerful models.

Model Size Limited by Tools

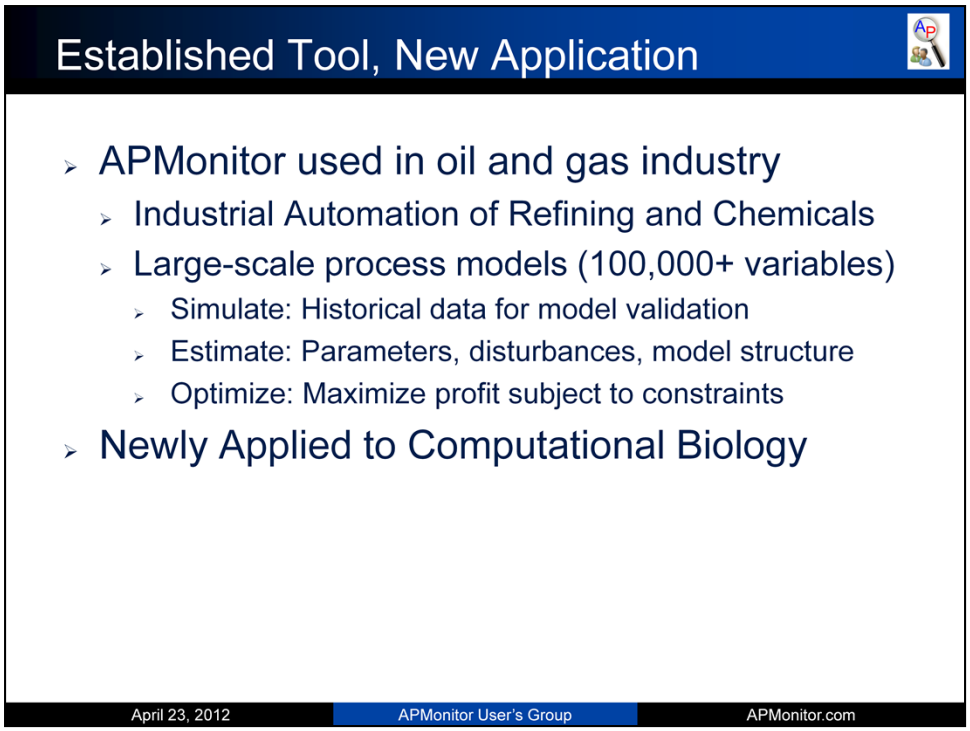


We need better tools (parameter estimation, optimization) to deal with large models!

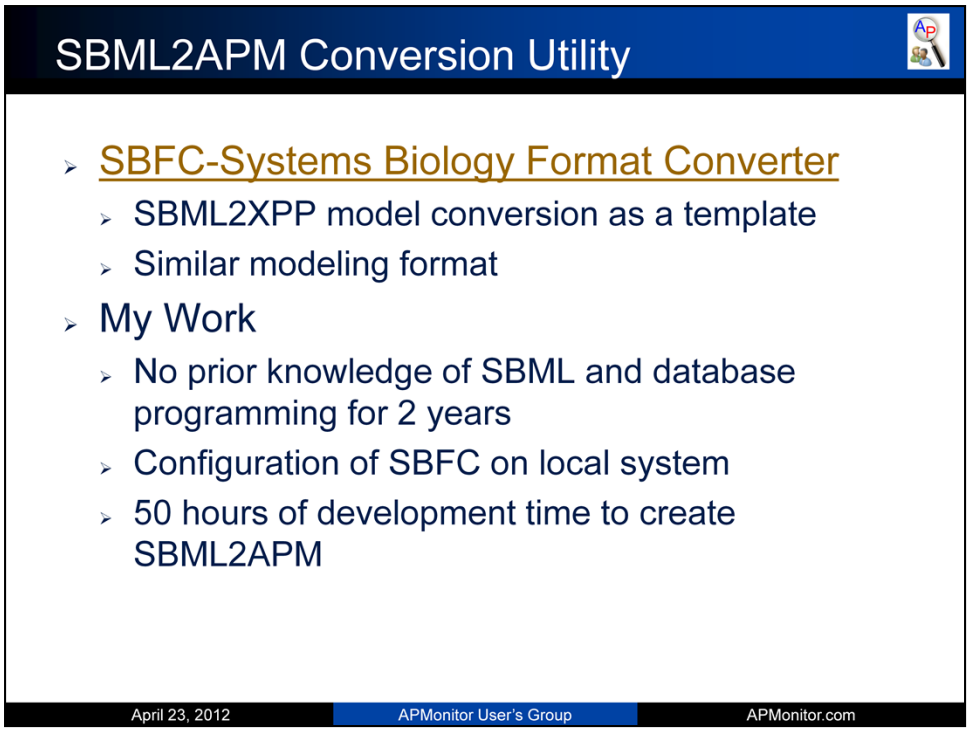
- Large ErbB signalling model (~504 physical entities)*
- Parameter estimation (simulated annealing) took "24 hours on a 100-node cluster computer"

*Chen et al. Mol Syst Biol. 2009;5:239 .

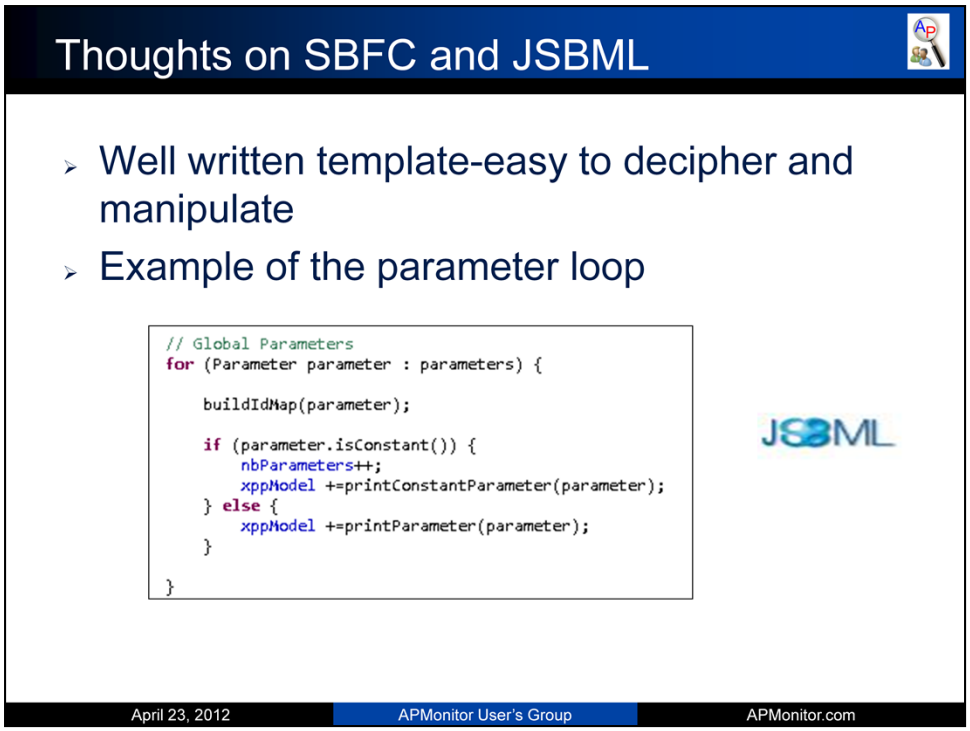
The second largest model in the biomodels database is the ErbB Model which contains 504 physical entities. It is a very complex model that took a very long time to solve using simulated annealing. Even using a very powerful computational system the parameter estimation took 24+ hours to converge to a solution for one parameter and with over 200 parameters only selected parameters could be optimized to fit the data. One thought to improve the efficiency for producing large scale models is to provide better tools which can handle thousands of variables.



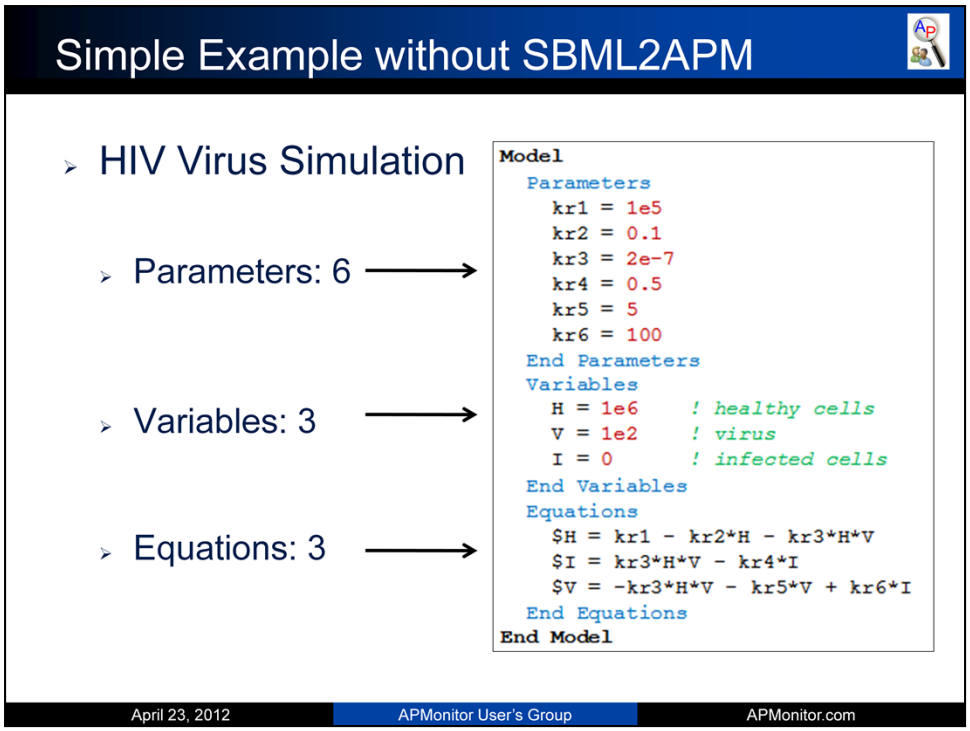
One tool that can do this is APMonitor. APMonitor is not a new software. It has been around for many years but has found itself mostly used in the oil and gas industry for the application of model predictive control using a simultaneous solving method with a specified historical time horizon in order to accurately predict the necessary present value needed for the control system. In order to perform this function it has been used to simulate, estimate, and optimize large scale dynamic models containing over 100,000 variables in a timely manner. Dr. Hedengren used to work in the oil and gas industry and has seen how this has worked first hand and Casey Abbott and I have been helping him to apply APMonitor to the models contained in the biomodels database and consequently Computation Biology.



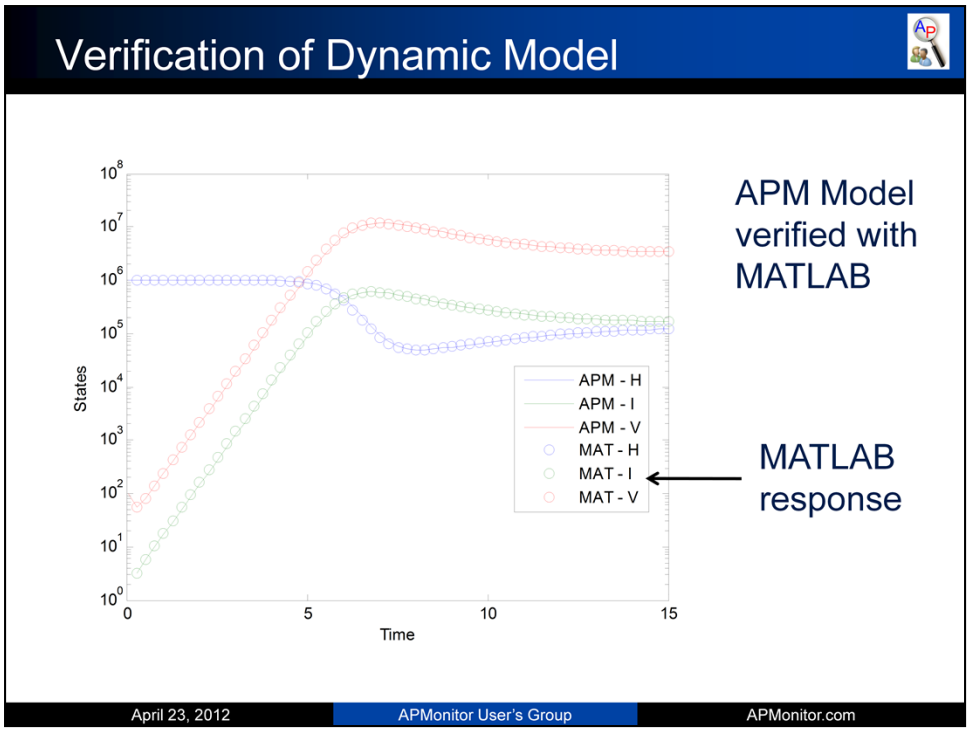
In order for APMonitor to reach a wider variety of models with ease a method to convert the models into a format APMonitor could handle had to be created. We looked into using a MATLAB converter and also a java based converter. After a couple of weeks determining which method would be best the java converter seemed much more promising since some effective tools for converting the SBML models had already been created. Namely the SBFC: Systems Biology Format Converter and its SBML2XPP conversion utility. This converter was chosen because it had a similar output to what APMonitor needed to run. This is where my main work for the semester began. I started in January with no prior knowledge of SBML or the biomodels database but with 2 years of VBA database programming experience and 2 java projects. The first big hurdle for me to develop the conversion tool was the setup of my development environment because it had been awhile since I had worked in Java. However, Nico Rodriguez from the biomodels database support team helped me get started with the open source code developed for the SBFC. Once I got started I was able to create a working version of a new addition to the SBFC: SBML2APM in approximately 50 development hours. My work was speeded up greatly with the way that the SBFC and JSBML libraries were created.



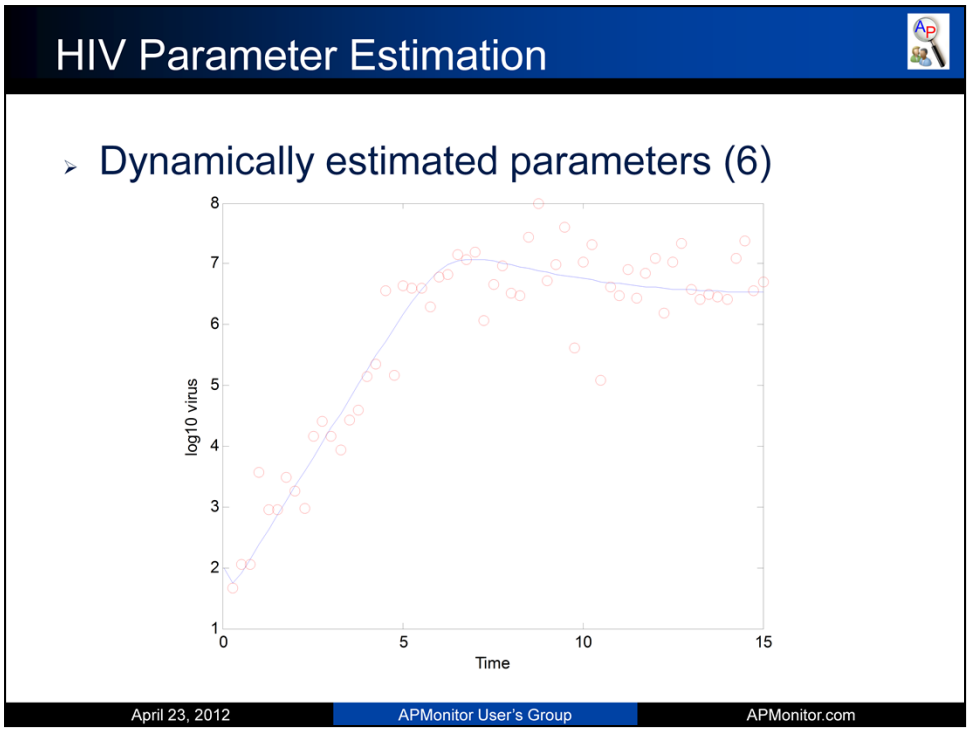
JSBML: the Java libraries designed to work the libSBML and the SBML files themselves to extract and convert data to desired format were essential in making SBML2APM tool possible. Dr. Dräger who is with us and his team developed this library and conversion utilities and I thank them for their work. The code was very well written and easy to interpret the functions and desired results. This made editing the SBML2XPP conversion utility to make what we needed much easier. Shown on this slide is a snippet of the code from the SBML2APM java file which searches through the parameters listed in the SBML file and prints the correct ones to the correct section.



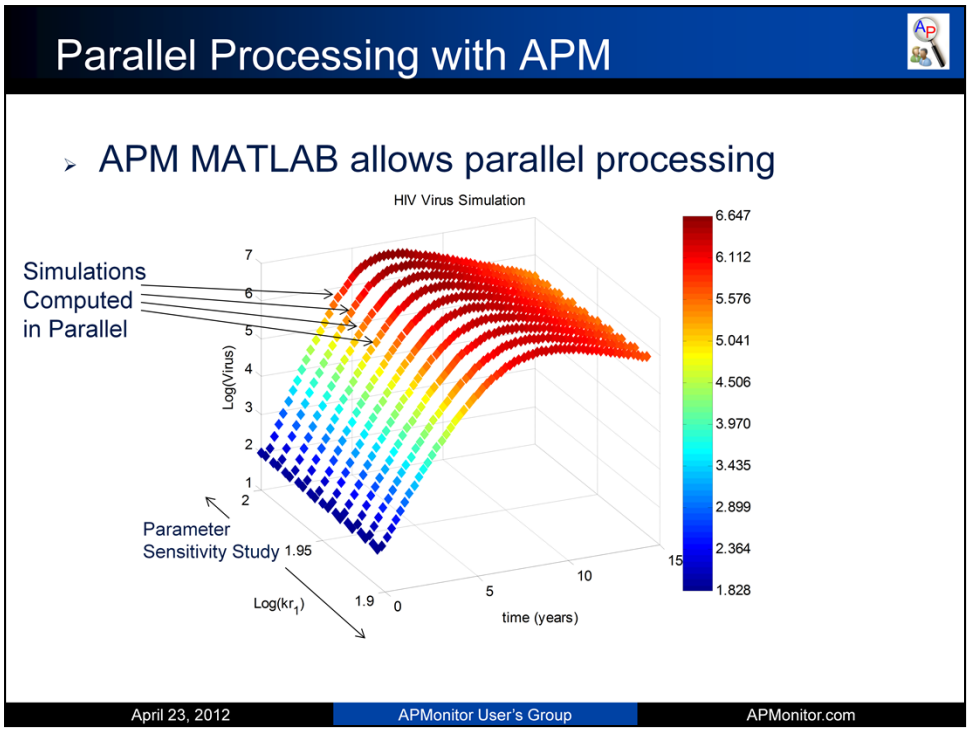
Here is a result from one of the HIV Virus models converted to the APMonitor format. APMonitor requires that the constants be declared in the parameters sections, initial values be declared in the variables section, and equations defined either as an intermediate calculation or in a state variable equations section. More on the different sections required by APM can be found on the APMonitor website. In this model we have a square model with 3 equations and 3 variables with initial values that are controlled by 6 parameters.



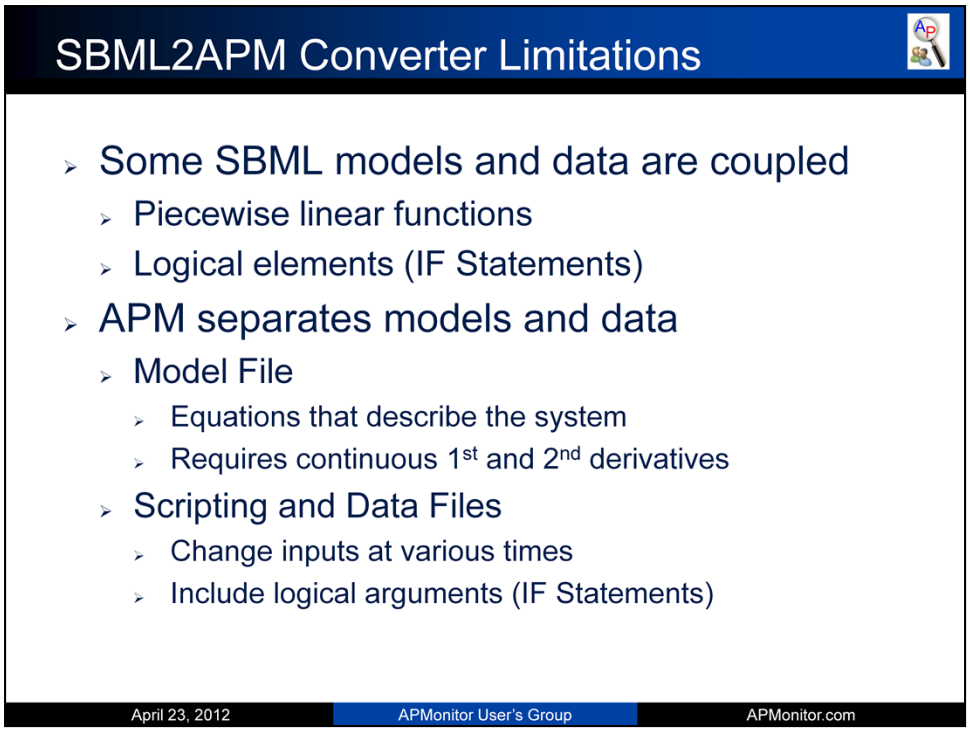
In order to validate APMonitor's simulation method we simulated the model in both MATLAB and APMonitor and compared the results. As you can see, the different software tools match up perfectly when simulating the model. However, simulation is not the only thing APMonitor is capable of. Remember those 6 parameters?



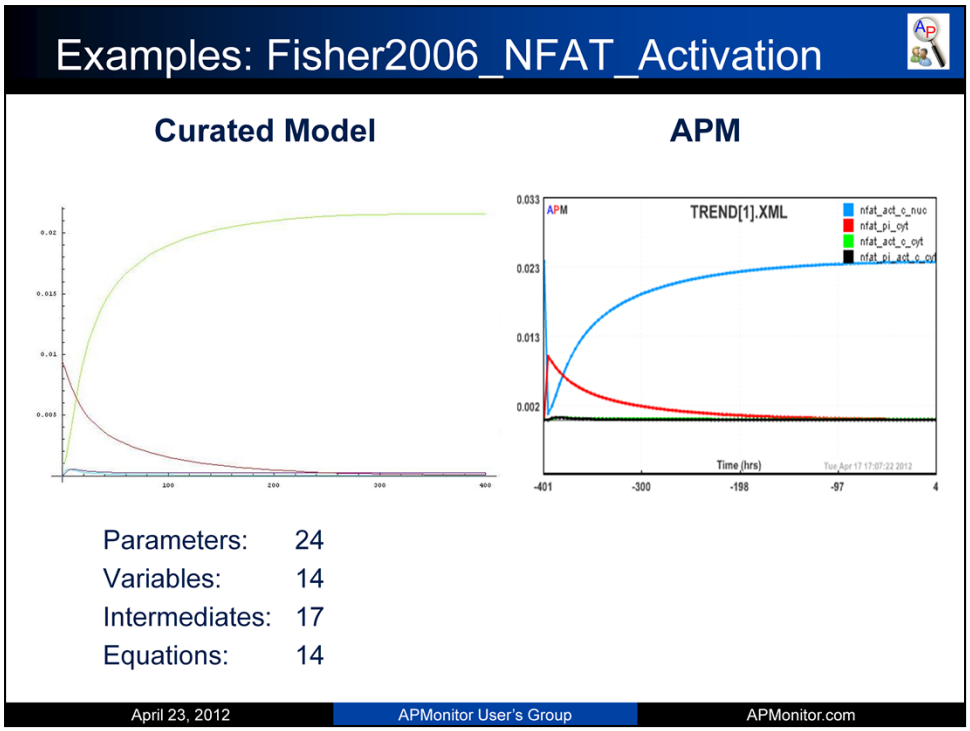
We simulated data from the model and then added noise to test APMonitors capability to estimate the parameters and reproduce the same desired model. As you can see, APMonitor was also able to estimate the correct parameters even after the noise had been added. We are aware that it is not something new for a software package to be able to correctly estimate parameters, but another benefit to the APMonitor parameter estimation is that it can also perform in parallel.



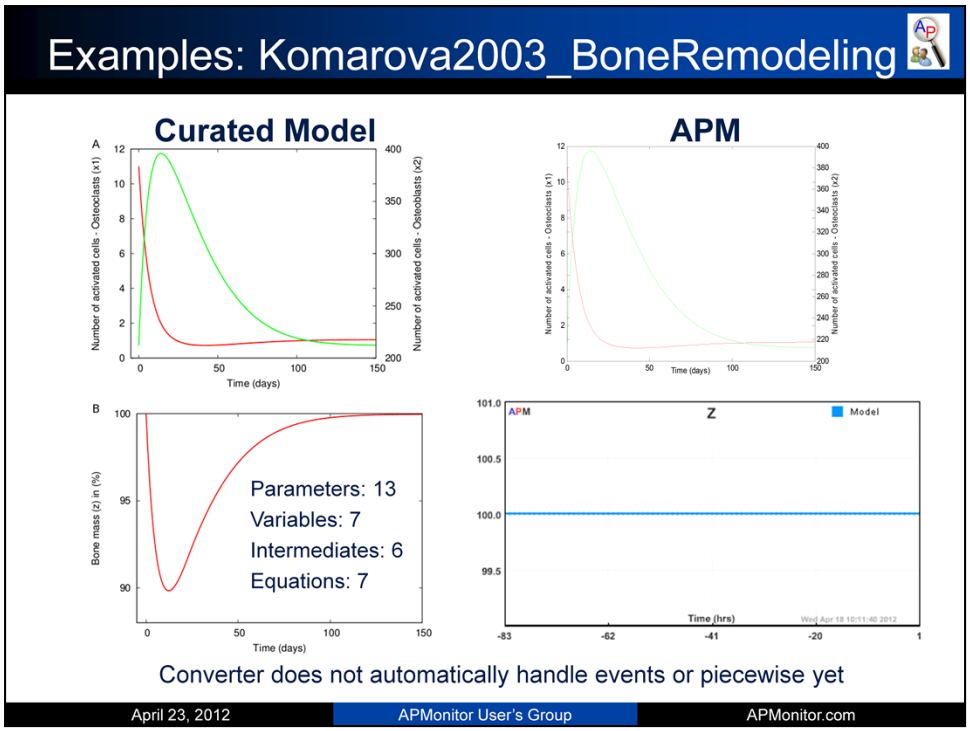
Using APM in conjunction with MATLAB allows for a parameter sensitivity study to be performed in parallel by running multiple simulations simultaneously. The benefit of this is drastically reduced solving times for large scale models.



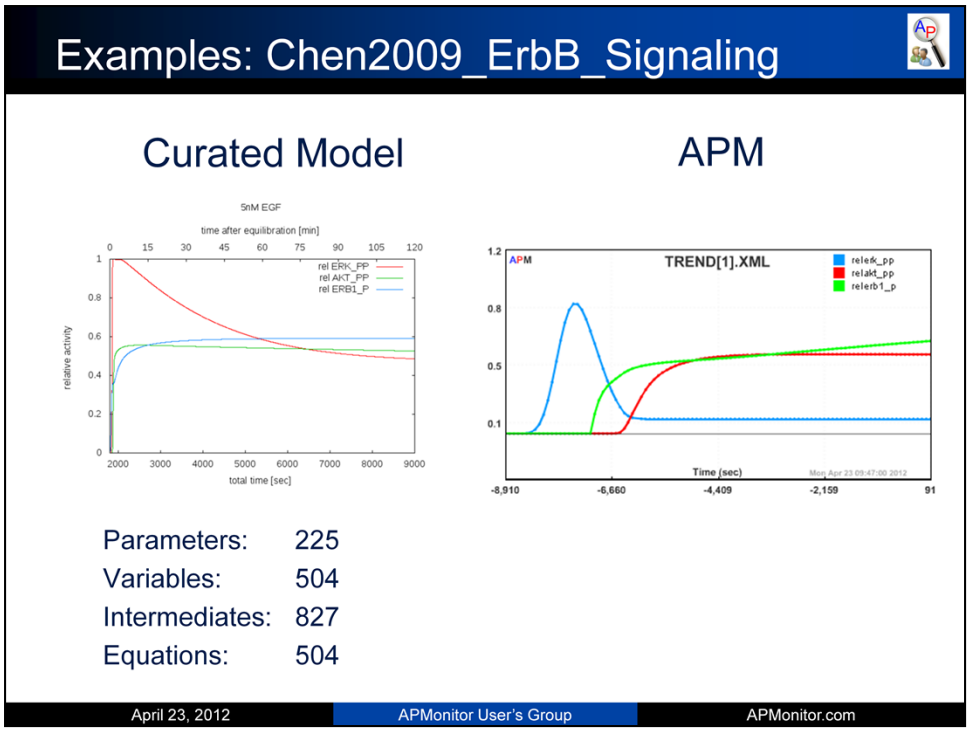
However, even with all that APMonitor can do the conversion utility is still not perfect, and part of it has to do with the manner in which the biological models are created. SBML allows for piecewise data changes and conditional events to cause changes to the model creating a system where the data and the model are coupled. Since APMonitor separates the data from the models this presents a challenge when performing an automated conversion. APMonitor can handle the logical if then statements and parameter changes, but when it presented separate as data and not as the actual model equations. Hopefully if APMonitor is accepted by the biological modeling community larger scale models can be produced that keep the model and data separate to allow for more efficient simulation and parameter estimation.



Still, we would like to show some examples using the SBML2APM converter tool. This NFAT_Activation model was download directly from the SBML database, converted to the APM format with SBML2APM and then simulated using a web interface. The time on the x-axis is arbitrary. As you can see the conversion utility works to simulate normal sized models from the biomodels database that do not contain piecewise logical statements or conditional events.



The next model, Bone Remodeling illustrates the limitation that I explained in the SBML2APM converter. The z for the bottom curated graph changes based on a conditional event in the concentration of x1 and x2. The next example will show one method of how to deal with this problem. The top right graph was created in MATLAB using the APMonitor simulation in order to present the same axis as the curated model for the values of x1 and x2. Again, when no piecewise or events are altering the model the conversion utility works to allow APMonitor to simulate the model correctly.



Our big challenge was in creating a working version of the ErbB Signaling model. It contains parameters that are altered by a conditional step function. Since the condition was time the parameters were separated from the model into a data file in which they were altered at the appropriate times to match the effect of the conditional step function. By doing this the model was then possible to be run using APMonitor and was simulated correctly very quickly considering the large scale of the model. Remember, this was the model presented in the beginning that took 24 hours to run one simulated annealing.

Computational Time in APM

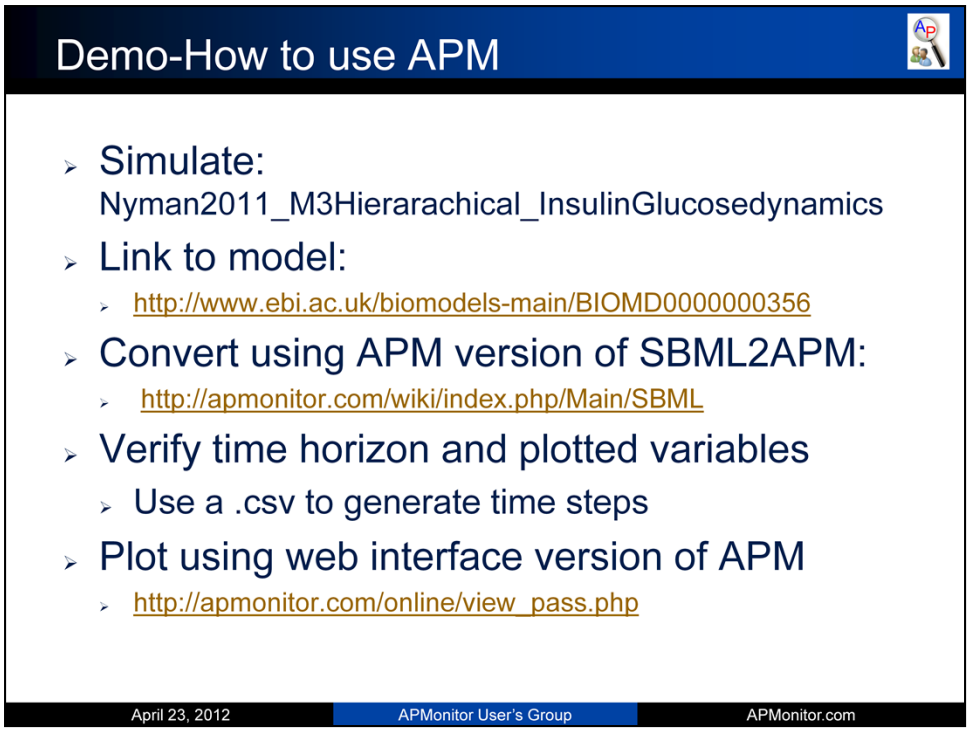
	Variables	Time Steps	CPU Time (sec)	CPU/Time Step
Bone Remodeling[1]	11	150	1.40	0.0093
NFAT Activation[1]	30	100	2.19	0.0219
Vaccination Invasion[1]	37	100	3.06	0.0306
ErbB[1]	504	100	54.70	0.5470
ErbB[2]	504	100	22.70	0.2270

1 = Windows 7, Intel i7, 8 CPU
7k RPM HD, gcc Compiler

2 = CentOS Linux, AMD, 64 CPU
15k RPM HD, Intel Compiler

April 23, 2012 | APMonitor User's Group | APMonitor.com

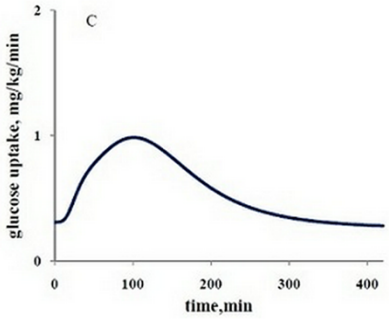
In APMonitor the ErbB model was able to be simulated on a CentOS Linux AMD, 64 CPU 15k RPM Hard drive in 22.7 seconds. On a much smaller system the model was still able to be simulated in less than a minute and the other smaller models in this presentation were completed in less than 5 seconds of computational time. Of these models, I have not presented the Vaccination Invasion model. That is because I wanted to perform this model simulation with you so you can see what I am talking about with using the conversion utility and then getting it to simulate using a web interface.



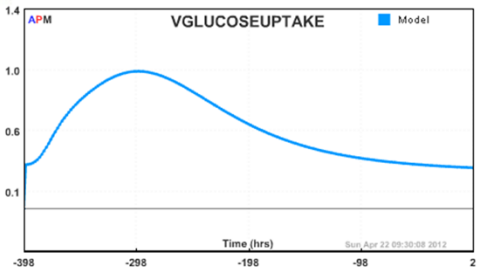
For this simulation we first need to download the SBML model file. After downloading we need to convert the file. Then, a key point in being able to recreate any model is understanding the time steps and plotted variables. The time steps are one limitation and advantage for APMonitor. APMonitor does not support adaptive time steps and I will explain that as we perform the demonstration. After verifying the time steps and the created the appropriate variables to plot the web interface can be used to perform the simulation.

Demo - Results

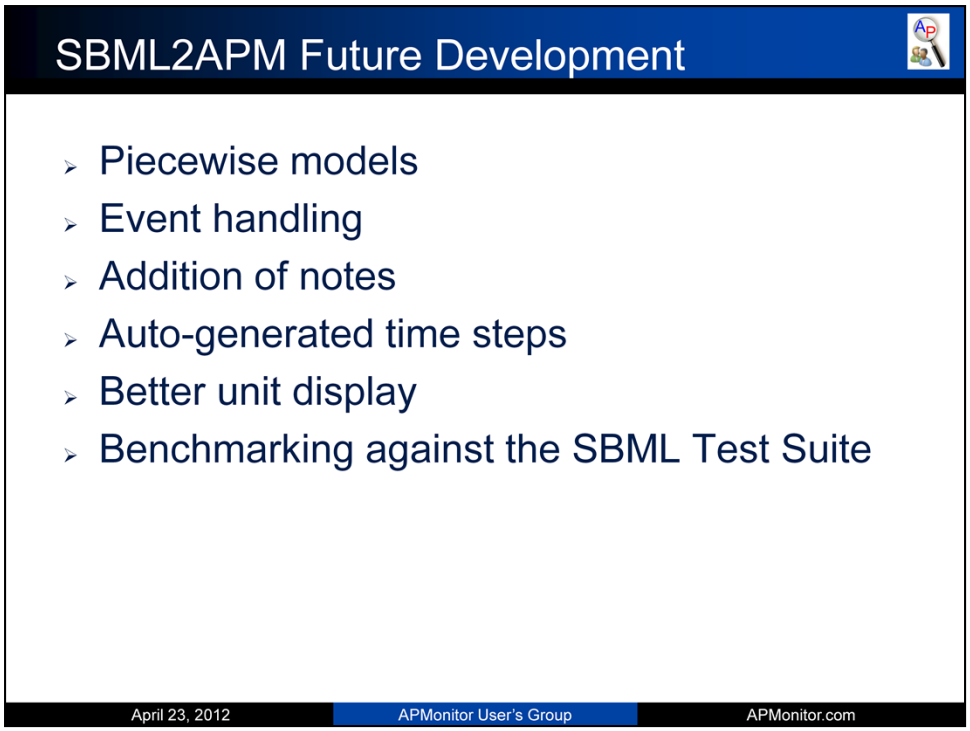
Curated Model



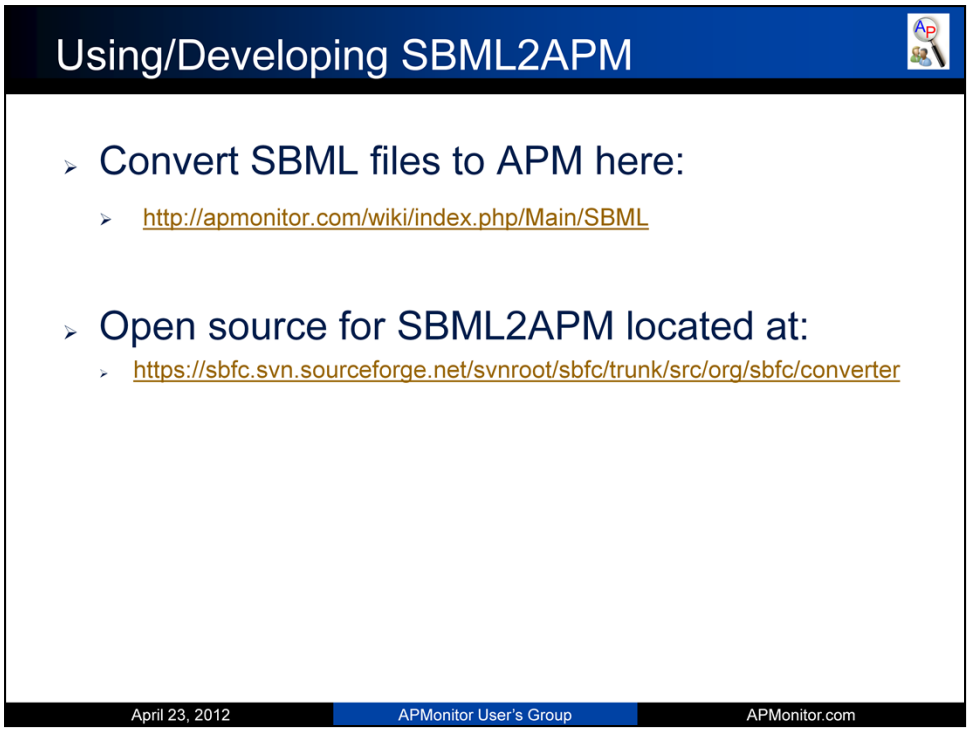
APM model



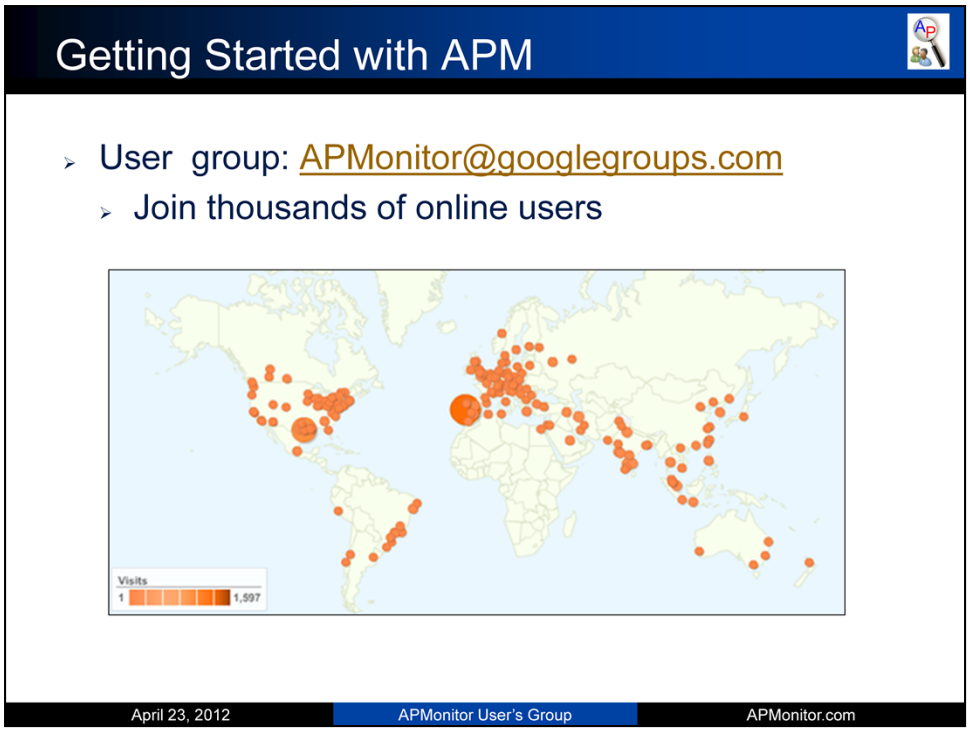
Parameters:	80
Variables:	67
Intermediates:	64
Equations:	67



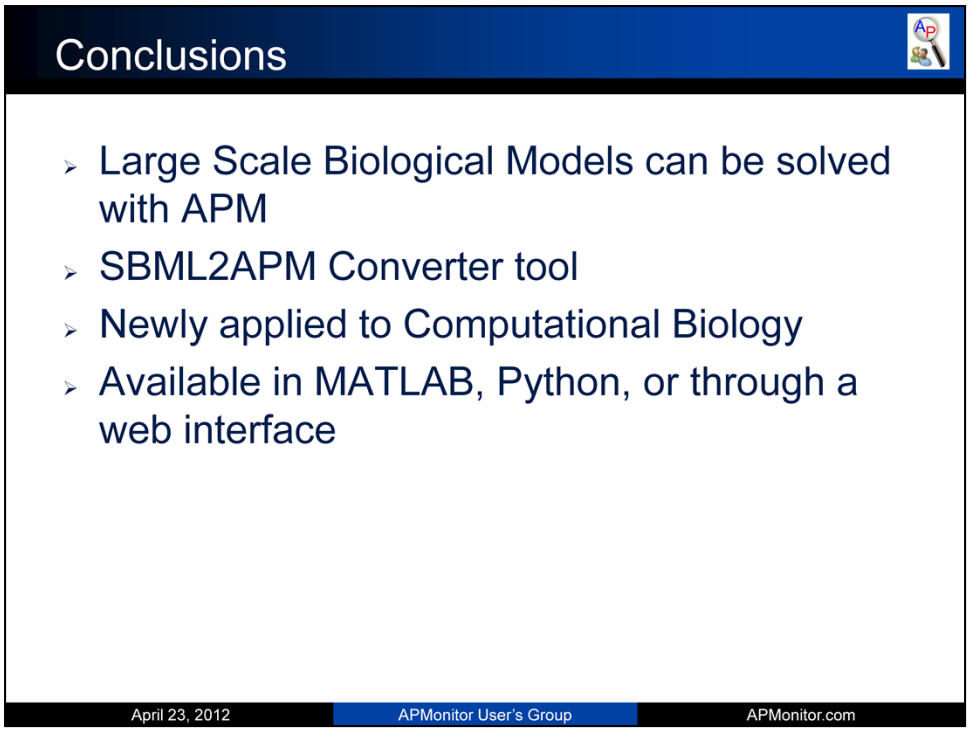
Some aspects of the conversion utility are still not yet complete, but part of it also has to do with the general incompatibility of data coupled with models for APMonitor. Verification that the conversion utility is working correctly on more models and feedback will also be helpful.



If you are interested in testing the conversion utility remember SBML files can be converted on the APMonitor website. If you are interested in looking at or assisting with the development of the source code it can be found on the sourceforge website containing the SBFC conversion utility.



For those wanting to use or test out APMonitor, don't feel like you have to figure it out on your own. There is a users group with thousands of participants around the globe who are constantly using APMonitor and can help you understand and resolve any issues you may come across when trying to get started or running your models.



In conclusion, Large Scale Biological Models can be solved with APMonitor very efficiently. The SBML2APM conversion utility is ready and accessible on the APMonitor.com website and will potentially be available on the biomodels database website by October. Even though this is a new application to computational biology it is a system that has been tested and proved effective as a means of large scale modeling and optimization in a different but also very complex system of reactions and models. As a final note, something that was not emphasized during the presentation but mentioned throughout is that the APMonitor software can be used via a MATLAB, Python or web interface depending on what your preferences as a user are.

Acknowledgements

- Vertex Collaboration
  - Large Scale Parameter Estimation
  - Identifying parameters to be estimated
- Casey Abbott
  - ErbB and HIV virus modeling applications
  - Parameter sensitivity studies
- Trevor Slade
  - MATLAB interfacing with APM/SBML
- Creators of the JSBML library and SBFC converter package
  - Andreas Dräger and Nico Rodriguez

References

- Chen et al. Mol Syst Biol. 2009;5:239 .
- Dräger A, Rodriguez N, Dumousseau M, Dörr A, Wrzodek C, Le Novère N, Zell A, and Hucka M. **JSBML: a flexible Java library for working with SBML**. *Bioinformatics* (2011), 27(15):2167–2168.

Additional Slides

Navigate to an Optimal Solution

$$\min_{x \in \Omega} \; J(x,u)$$
$$s.t. \;\; 0 = f(\dot{x},x,u)$$
$$0 = g(x,u)$$
$$h(x,u) \geq 0$$

Characteristics
- Nonlinear
- Multiple local minima
- Mixed Integer
- Large-scale

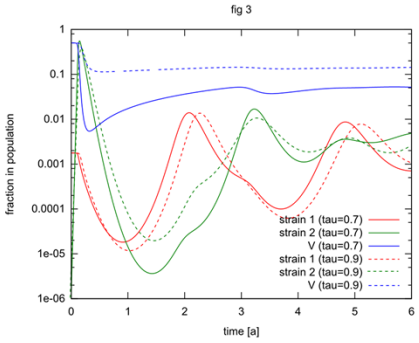
Methods
- Interior Point
- Active Set
- Global Search
- Parallel Computation
- Branch and Bound

Survey of DAE Solvers

Software Package	Max DAE Index	Form	Adaptive Time Step	Sparse	Partial-DAEs	Simultaneous Estimation / Optimization
APMonitor	3+	Open	No	Yes	No	Yes
DASPK / CVODE	2	Open	Yes	No	No	No
gProms	1 (3+ with transforms)	Open	Yes	Yes	Yes	No
MATLAB	1	Semi-explicit	Yes	No	No	No
Modelica	1	Open	Yes	Yes	No	No

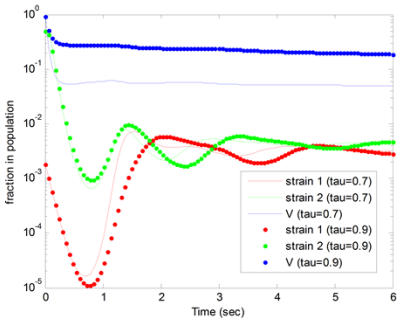
Demo - Restif2007_Vaccination_Invasion

Curated Model



Differences in initial conditions

APM



Parameters:	12
Variables:	23
Intermediates:	26
Equations:	23