CAREER ABI Synopsis: Modeling made easy

The research goal is to extend rigorous mathematical simulation methods and make them available in a user-friendly model description language designed to solve computationally challenging forward simulation problems in biology. Many user-friendly simulation tools have been developed in systems biology, but their underlying mathematical representation impairs their use for important problems in genetics and ecology. The proposed work will fill that gap. It will transform quantitative modeling into a mainstream activity that will be an integral part of biological research in the future.
Intellectual Merit: Quantitative models can help organize, develop, and test our thoughts about complex problems, yet many biologists refrain from using them. Aim 1 is to implement a new language that makes it easy to read and write such models for humans and for computers. To harness the large-scale computing power needed for analyzing complicated models, a link will be built to the evolution@home public global computing system and to private Condor pools. To efficiently store the results produced, a new standard for storing and sharing simulation results will be pioneered. Aim 2 is to expand systems biology approaches to tame combinatorial explosions in genetics and ecology models. A combinatorial explosion occurs if a simulation has many more potential types of parts than actual parts. Such combinatorial explosions have been explicitly addressed by simulation tools in systems biology, but are fundamentally different from those in genetics. Aim 2 will develop algorithms and representations that work efficiently with both, and that also allow for distributions of event times other than the standard assumption ("exponential"). Aim 3 applies these new tools to intensely studied biological systems, including growth of VSV viruses in cells and adaptive evolutionary ecology of copepods (important zooplankton food source for fish). Close collaborations with neighboring labs will help shape the development of tools to ensure the resulting cyber-infrastructure is useful for cutting edge research. Such research is pivotal for increasing the quantitative rigor, realism and accuracy represented in models that will become increasingly important for various decision support systems.
Broader Impacts: The same language and simulation tools developed for research will be used to teach graduate and undergraduate students, and design innovative teaching materials that improve the understanding of quantitative modeling for diverse audiences. Using 'the real thing' is fascinating and transformational for learning, as witnessed by the rise of "R" in statistics. Working with K12 teachers, materials will be developed that explain the importance of good models to a broad audience. To this end, a brief interactive course with comprehension tests will be implemented to impart basic modeling knowledge, encourage responsible use of models, and discourage abuse. Successful completion of the course will provide a "License for Using Models". This and other teaching tools will be developed and tested on K12 students and evolution@home participants. The latter contribute CPU power to simulations of evolution and have a natural interest in models they are simulating, an interest to be met by an engaging website. The overall vision is to raise awareness for the importance of good quantitative models. The PI's career is to build such models.

CAREER ABI – Modeling Made Easy: Extending systems biology modeling approaches to genetics and ecology

Context and overarching aims

The research objective of this proposal is to make rigorous simulation methods from computational systems biology more accessible to biologists and apply them to modeling problems in systems biology, genetics, ecology and evolution.

Aim 1 will transform how biologists view models by greatly reducing the time needed for building and analyzing models. This is critical for my long-term aims of building more realistic models of evolution and for estimating important poorly known evolutionary parameters with the help of systems biology models (e.g. epistasis, mutational effects [1]). Analyzing realistic models is often computationally challenging, so integrated support for distributed computing is critical.

Aim 2 will bring the ease of modeling known from systems biology to genetics and ecology. This is achieved by developing new mathematical algorithms for taming the combinatorial explosions caused by evolutionary processes like selection and recombination. The different nature of combinatorial explosions prevents current systems biology modeling approaches from being effective tools for evolutionary questions. Aim 2 will change that. Aim 3 will apply this system to real-world examples.

Simulation methods are becoming more important in modern biology, as they can help analyze systems that cannot be observed experimentally and that are too complex to be solved by analytical mathematics [1][2][3]. This view is shared by many researchers who have relied on simulations for analyzing the systems they study.

Yet many biologists are skeptical of simulation work. Controlling quality before accepting models is pivotal [4][5], as common problems abound: (i) the absence of general and rigorous tools forces costly re-implementations, reducing time invested in biological aspects of modeling; (ii) lack of computing power and poor parameter estimation reduces the reliability of conclusions; (iii) difficulties with math and computing reduce the adoption of quantitative models in biology.

My overarching aim for the proposed work is to help reduce the barriers for high-quality modeling work. I will do this by developing a user-friendly model description language, and a toolkit that is easy to use for a broad range of modeling problems in systems biology, genetics, ecology, and evolution. Here I call this language and the related toolkit "ε" for brevity. ε shall encourage good modeling practices and will come with support for distributed computing and parameter estimation. ε will help transform how biologists think about models by its:

- Speed and ease of implementation for models from a broad range of domains,
- Conciseness and quality of model documentation,
- Integration of research and teaching features.

ε will serve as an important foundation for my research for many years to come. To ensure ε becomes increasingly useful for research, a broad range of modeling topics from systems biology, genetics, ecology and evolution will be addressed as part of the proposed work. I bring a rare combination of skills and motivations to the development

of ε: I am motivated by biological questions to get real biology done, yet I am fluent enough in math and computer science to lead the development of good algorithms and state-of-the-art tools. The proposed work can be cut in small enough pieces to guarantee important progress on an annual basis. At the same time, the work of increasing modeling capabilities, improving simulation algorithms, refining parameter estimation techniques, and using all this to analyze interesting biological questions in realistic models is in no danger of running out of fascinating questions any time soon.

ε will also serve as an important foundation for my teaching. One explicit development goal of ε is ease of use. Using ε in a teaching context is one of the best tests for this. My integrated approach is designed to overcome the complications that arise from separating 'teaching toys' lacking the power for serious research from 'research tools' lacking the ease of use that might encourage beginners to start using them. By actually using ε for both research and teaching, researchers and students will win increased productivity and increased research power, respectively. Using ε in a broader outreach context will help communicate some of the excitement of research and hopefully inspire a new generation of students.

Designing such a large software system is what software engineers call a "wicked problem" [6]. In practice such problems are solved by incremental, iterative approaches, as new 'solutions' reveal previously unknown limitations. How can I be sure, ε will not become yet another big, failed IT project? The answer is two-fold. First, extraordinarily short 'communication lines' between the lead biologist using this (me) and the lead system architect (me) make development more efficient. This interdisciplinary extends to my group (currently a mix from computer science, math, biological and chemical engineering and biology). It means we implement what we need for biological research. Achieving this in a non-biology department can be difficult. Simultaneously, our close ties to computer science, math and statistics provide the expertise to pick best-of-breed solutions, which can be difficult in a biology-only department. Second, we will expand the system incrementally by following the "development cycle" outlined in Fig 1 and by using project management best practices.

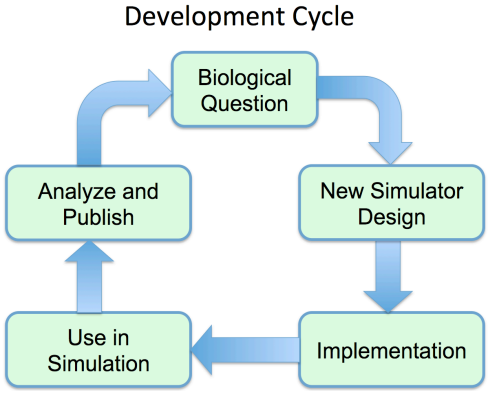

***Fig 1**: Development cycle for work on ε. Each new cycle starts with a biological research question that requires a new feature for ε. This feature is designed, implemented, tested and used for simulation work addressing that question. Results are then published before next round's design. This development cycle ensures a balance between biological research and programming work.*

I have gone through several related development cycles and expect the work to progress in more such cycles. The vision behind my work is to create a model description language that does for forward simulations in biology

what 'R' did for statistics: produce an open-source tool that is expandable and useful for a growing range of problems.

The remainder of this proposal is structured as follows. Aim 1 targets basic aspects of ε and the toolkit's software infrastructure. Aim 2 extends models beyond the limitations of current computational systems biology modeling frameworks in order to apply them to genetic, ecological, and evolutionary problems. Aim 3 concentrates on specific research problems from neighboring labs that I plan to model using the new extensions of ε. The broader impact section elaborates on the integration of education and research facilitated by ε and on outreach activities highlighting the importance of modeling and of evolution for addressing challenges we face in our world today.

Aim 1. Combine basic ε with global computing and parameter estimation

Background: Separating model description from mathematical analyses

One of the elegant results of formal modeling research in computational systems biology and other areas is the demonstration that models can be encoded independent of the mathematical analysis techniques used to analyze them later [7] [8] [9] [10] [11] [12]. To facilitate corresponding mathematical analyses, the model is automatically transformed into appropriate data structures and equations (Fig 2).

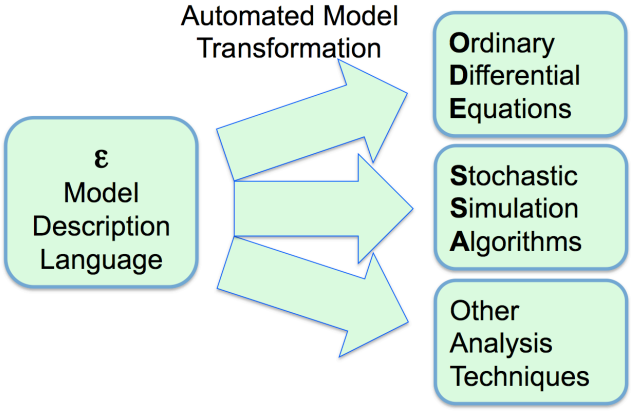

***Fig 2**: Automated model transformations into different mathematical analysis techniques facilitate more flexibility in choosing appropriate techniques.*

Such automated transformations help exploit the strengths and avoid the pitfalls of different mathematical analysis techniques. For example, Ordinary Differential Equation solvers (ODEs) are very efficient in analyzing systems with particle numbers large enough to render the randomness of stochastic noise irrelevant [3]. However, ODEs should not be used when fractions of particles start to matter. In such systems ODEs will give meaningless answers as using them can, for example, imply the assumption that 0.8 molecules take part in a given reaction. Such systems have to be analyzed with Stochastic Simulation Algorithms (SSAs), which respect the fundamental integrity of molecules and individuals [3]. Even if the application of both techniques can be justified, one might still want to use them both: ODEs are efficient at computing precise expectations, but their deterministic nature does not provide estimates of variability. SSAs provide good measures of variability by computing many stochastic runs, but are much slower than ODEs.

In the absence of automated model transformations, researchers have to encode their model either directly in ODEs or they have to implement a stochastic simulation. The corresponding manual transformations are cumbersome and known to be error

prone. Hence automated transformations of models substantially boost model implementation speed and power for analysis. I have worked before with the Bio-PEPA process algebra [8] [9] [10] [11] [12], which is such an abstract modeling system that allows for automated model transformations. I have seen the nature and elegance of model transformations, which usually happen at the click of a button. Research in computational systems biology seeks to extend and develop new automated analysis techniques that answer specific questions about a model, sometimes even without the need for simulating it (e.g. [8] [13] [14]). Hence, transformations to "other analysis techniques" are included in Fig 2, highlighting my interest to include useful analyses that come out of such research into ε. Unfortunately, most biologists have difficulties with writing process algebra code.

Analyzing stochastic and deterministic versions of the same model is a recurrent theme, not only in systems biology, but also in genetics, ecology and evolution. Thus I will work towards developing the capability to automatically transform abstract ε models into both stochastic and deterministic model versions.

Goal 1.1. Develop ε and its simulator for user-friendly modeling

Many abstract model description systems have been devised over the years (e.g., see [15] [16] [17] [8][18][19] [20] [21] [22]), some more intuitive to use than others. Popular markup languages such as SBML are powerful abstract model descriptions [23], however the syntactic clutter of XML makes them very cumbersome to use without additional layers of tools. Also, systems biology simulation algorithms make assumptions different from those needed for simulating genetic systems (see below). There is no language for describing biological systems that is as elegant and widespread as "R" is for statistical problems.

Modern compiler construction tools like ANTLR [24] make it easy to build domain specific languages that can be specifically designed for concise implementations of particular problem types [25]. I have used ANTLR before [26][10,11] and propose here to select and implement the most elegant model description formalisms.

For example, after exploring several formalisms, I found a pattern that consistently showed excellent usability for biologists: Describing systems in the form of many chemical reactions is very intuitive for biologists as they frequently use equivalents of this formalism to represent structural information about their models. I made this formalism a central feature of ε, but generalized it to fit anything from biochemistry to ecology equally well. Instead of "molecules" or (molecular) "species" and their "reactions" an ε model consists of "parts" and their "actions". Let's examine a simple example of ε code: The first two lines represent reaction $A + B \longrightarrow C$ with default kinetics "MassAction" occurring at default rate "1" and at the non-standard rate *r* = 0.5:

```
action { A + B  ---> C } // no need to specify defaults
action { A + B  ---[r = 0.5]---> C } // rate r = 0.5
action { S<sub> ---[  E<enz km=0.5 vm=50>
                    - law MichaelisMenten ]--->  P  }
```

The Michaelis-Menten reaction $S \xrightarrow{E} P$ requires specifying a kinetic law. The prominent feature of the arrow '`--[ - law ]-->`' serves as a flexible organizing principle for separating the parts involved in this action. Having '`law`' at the center reminds biologists of the important role probabilistic laws play in governing the system. However, only the name of a law is given here. The input parameters needed to create an instance of the law for this particular action can be distributed throughout the action and are marked by '<…>'. A law is defined only once by listing its input parameters and the mathematical equation at a location where users are not likely to change it. When the parser finds a reference to a law within an action, it searches for all required input parameters in the angular brackets '<…>' of this action and then instantiates it as if the equation was written for this particular action. Other languages require the user to do this work, resulting in large numbers of similar equations that are hard to read and easy to confuse as they usually differ only by some numbers. Such numbers have no particular meaning, are hence difficult to debug and can be considered the "goto" equivalent for systems biology models (i.e. a programming construct that is error prone). To my knowledge, I pioneered a precursor of the '`law`' construct presented above [11,26]. It has the potential to substantially simplify modeling. I am currently in the middle of implementing the syntax given above. I have discussed this feature here to provide a flavor for the types of constructs that I plan to build into ε. During my modeling work over the years I have collected too many similarly useful language design ideas to describe here. My use of ANTLR to define ε, provides much flexibility and shifts efforts from implementing a parser towards focusing on desirable language features.

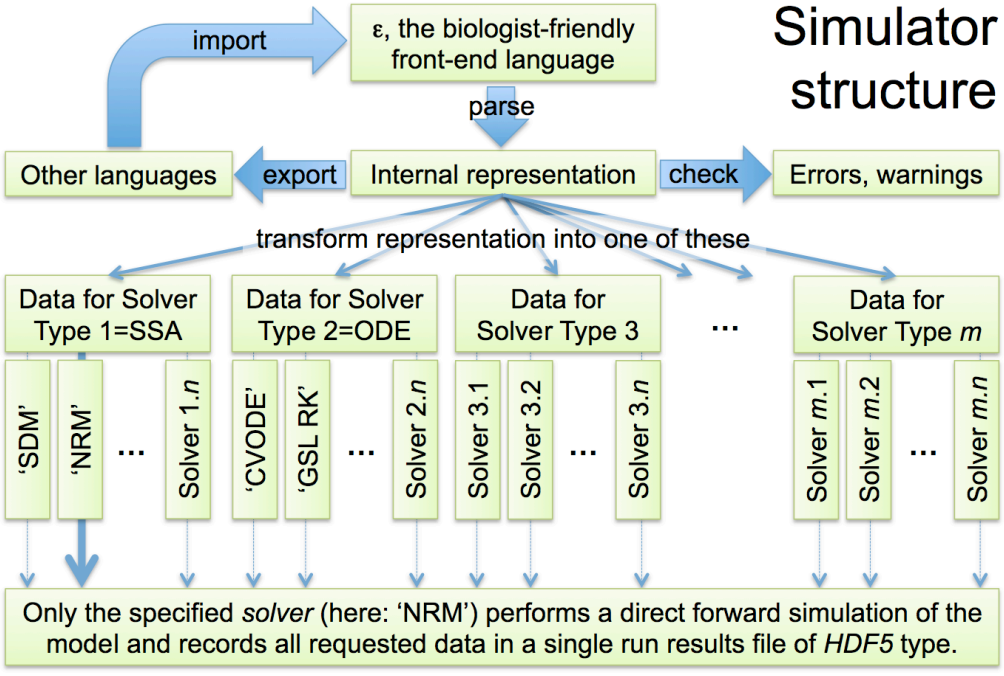

***Fig 3**: Overview of architecture and information flow of the ε simulator. Expanding 'Solver Type' and 'Solver' collections can provide simulation solutions for increasing ranges of problems making this infrastructure a safe choice for the future. Independent solvers can be used to test each other's accuracy. The thick arrow indicates which solver was selected in a hypothetical run ("NRM"). See text for abbreviations.*

As shown in Fig 3, once an ε model has been parsed its various fragments are sorted into an internal representation in C++. This helps reorganize the model into the form required by the solver type (e.g. SSA) requested for computation of the model. Then the actual solver requested in ε can start simulating the model and recording desired output. As shown in Fig 3 there are vast possibilities for developing different solver types and solvers in this infrastructure. I intend to support a collection of best-of-breed algorithms to be at the disposal of everybody for analyzing ε models. Examples include CVODES [27] [28] [29], the Runge-Kutta method from the Gnu Scientific Library 'GSL RK', the Sorting Direct Method 'SDM' [30], the Next Reaction Method 'NRM' [31], the Midpoint method [32], tau-leaping based on the midpoint method [33,34], the rule-based NFsim package [35] and others as they become available and as time allows to develop them in collaboration with David Anderson (UW-Madison). More solvers increase the flexibility of a biologist in choosing an appropriate analysis method. They also provide an excellent opportunity for testing the accuracy of solvers by computing a given model independently through different solvers and comparing results.

Due to the diversity of biological modeling problems I plan to support features in ε that will not be supported by all solvers. In this case the simulator will check whether the requested solver can indeed handle the problem. If not, an error will be reported. Similarly, automated checks that trigger warnings or errors will be implemented for many other potential modeling problems. To increase interoperability and to facilitate analyses that might be implemented in other modeling systems, I plan to support export and import into other modeling formalisms, whenever there is a need for it (but this is not at the core of the work presented here).

Various design choices will be reviewed extensively within and outside of the developer group by experts and by beginners, all helping to converge on the simplest possible alternatives without oversimplifying.

Deliverable: a formal language with appropriate levels of abstraction that makes it easy to write well-documented models with few opportunities for typos, even if model sizes become large, and a simulator that can simulate models described in that language.

Goal 1.2. Support distributed computing and parameter estimation in ε

Biological models do not respect our computational complexity categories: Potentially interesting simulations in disciplines from systems biology to evolutionary biology can take any time from seconds to years [36-38]. To support the rigorous analysis of more complicated systems, I will support the possibility for different parameter combinations of ε models to be computed across a distributed computing infrastructure. This exploits the already parallel nature of this computing task and can support parameter sensitivity grid searches as well as parameter estimation. Figure 4 provides an overview of the system that I plan to develop. I will leverage local collaborations with the developers of Condor [39] to develop the corresponding distributed computing infrastructure. Condor serves as an excellent resource-request-match-broker, once an appropriate description of the tasks is provided. Likewise, results need to be stored, exported, and plotted.

These parts of the infrastructure will be developed as needed. Simulation results will be stored in the HDF5 file format [40] and will be made available upon request, provided the means for transporting the data are available. Future work will explore different options for how to best share results. The possibility will be maintained to export simulation data as "tab table files", a format that is ready for importing and plotting in external packages (e.g. R, Excel). Integrated plotting solutions will be provided for quick browsing of large collections of results and for 'live' viewing of simulation data while it is being computed on the local machine.

I will also integrate this work with an ongoing pilot project in my group that implements parameter estimation in a systems biology model of cholesterol biosynthesis through Approximate Bayesian Computation [41-43]. The goal is to eventually support parameter estimation as an integral part of ε.

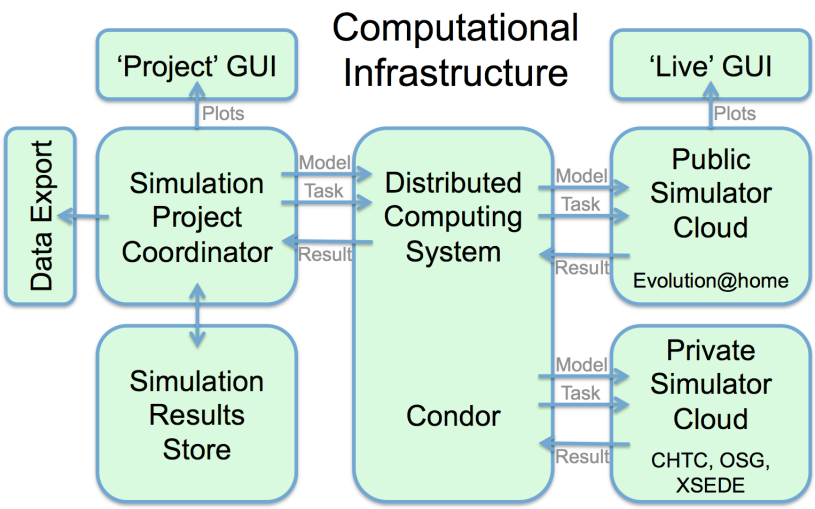

***Figure 4**: Computational infrastructure with most important information flows. GUI = Graphical User Interface, here mostly for plotting results. 'Clouds' are large collections of distributed computing workers that can compute tasks. CHTC = Center for High Throughput Computing, OSG = Open Science Grid, XSEDE = new NSF computing infrastructure.*

To provide the computing power needed for analyzing ε models, I plan to further develop "evolution@home", the first global computing system for evolutionary biology, which I started in 2001. I have been collecting ideas for making it an exciting hub that attracts computing power for simulation problems [36,38], while at the same time maximizing outreach potential (see Goal E.1). Such work will leverage substantial amounts of computing power from global computing "user groups".

Besides the public evolution@home 'compute cloud' I will also develop a link to private compute clouds, leveraging local CPUs at UW-Madison (CHTC, see letter of collaboration), remote CPUs at the Open-Science-Grid and the new NSF XSEDE facilities where needed. These will serve mainly two purposes: (i) recompute particularly important results on controlled resources if additional assurance of correctness is needed and (ii) computation of tasks that cannot be simulated on public resources because of their complexity (e.g. demand of too much RAM or bandwidth).

Deliverables: A working infrastructure that can schedule computing tasks on a public simulator cloud ('evolution@home') and on private simulator clouds ('CHTC, OSG').

Goal 1.3. Develop an HDF5 standard format for storing simulation results
When developing a general simulation system, some thought is needed about where and how to store simulation results, especially with a view towards long-term data management. While *ad hoc* solutions are easy to implement, they quickly degenerate into unmaintainable diversity, as different models lead to different formats and the meaning of particular values becomes difficult to determine due to lack of documentation. While self-describing XML may appear to be the answer to the latter problem, earlier pilot experiments I conducted led to the conclusion that XML does not scale nearly well enough to handle large amounts of data. A substantial search I conducted led me to choose 'HDF5' as my file format for storing simulation results [40]. HDF5 was developed for NASA for efficient long-term storage of satellite data and has recently started to attract biologists (e.g. [45][47]). It brings an enormous flexibility, speed and self-describing capabilities to the task. This flexibility makes the HDF5 programming interface more complicated than most users tolerate. Thus a number of file formats have been developed that are essentially HDF5 files but with additional constraints about how to store particular data. This allows the resulting user interfaces to be much simplified (e.g. SDCubes for collection of data arrays like visualization data generated in experiments [45], H5hut for time series of particle based simulations [46], BioHDF for next generation sequencing data [47], HDFEOS for Earth Observing System Satellites [48], PyTables [49], etc. For more, see HDF5 users in [40]). Storing big tables of scalar values is the easiest possible way of storing simulation results in HDF5 (similar to PyTables [49]). However, a more suitable data format for single simulation run results might allow for more flexibility to include different types of time-series as well as the recording of various types of snapshots (Fig 5). Storing research simulation results data on a large scale will benefit from the development of a consistent HDF5-based file-format for ε. Such a format does not seem to exist for simulation results of the structure presented in Fig 5. Our design will follow existing formats where reasonably possible to facilitate using external visualization solutions (e.g. PyTables in combination with SciPy). When developing such a file-format we will aim to make this as generally useful as possible.
Deliverables: A specification and a programming interface in an open source library for writing and reading a HDF5 based file format for storing summaries from forward simulation data. Functionality to export and plot data from this HDF5 format will be documented or developed and maintained.

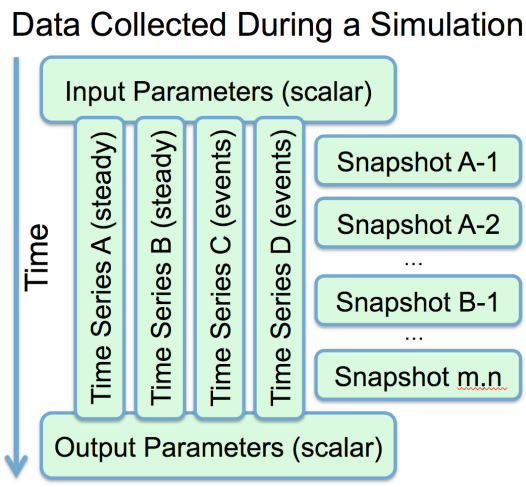

Fig 5*: Types of data that can potentially be collected during a single simulation run and that users might want to store in a HDF5 results file for a single simulation run.*

Aim 2. Extending modeling formalisms to support genetics

Background: Systems biology models with combinatorial explosions

There are many tools that simulate what I call here an "explicit biochemical reaction network" (e.g. see refs in [8][10]). All reactions in such networks have well defined input parts, well defined output parts and a reaction rate. If such a system is stochastic, then variability can only come from differences in the timing of the reaction (Fig 6A).

Some researchers have pointed out that even modest numbers of phosphorylation sites in proteins generate a combinatorial explosion that quickly makes it impossible to generate the explicit network, leaving individual-based simulations as the only alternative for analyzing such systems [20][50][51][35]. In this case, reactions are described as "rule-based biochemical reactions" (Fig 6B). They come with some uncertainty about the "reactant" input of the action. This uncertainty can only be resolved at runtime when the actual number of parts existing in the system is finally known. I plan to integrate the open source C++ NFsim simulator [35] into the ε system to simulate such models.

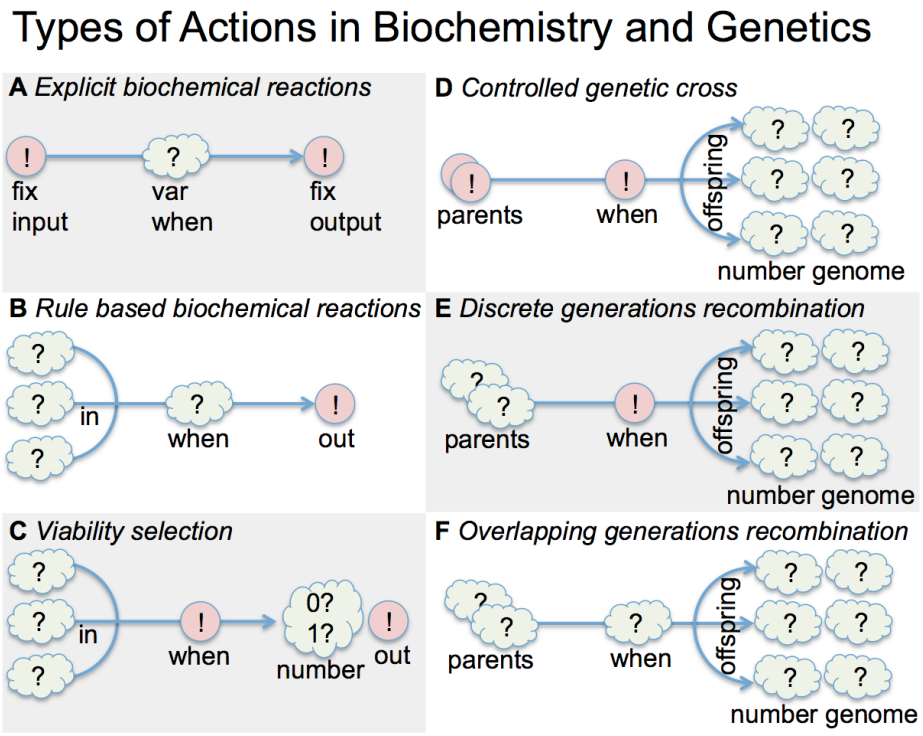

Fig 6: *Types of actions that ε shall support. Known quantities of an action are marked with "!", whereas "?" denotes potential variability due to randomness in the system.* ***A, B****: Biochemistry knows variation in timing and in input, but not in output.* ***D, E, F****: Recombination and selection in genetics introduce output variability.*

Goal 2.1. Systems biology models with arbitrarily distributed event times

Since Gibson-Bruck introduced the NextReactionMethod (NRM) [31], the simulation of arbitrary, non-exponential distributions of event times is possible in principle. Yet only few systems biology packages support this (e.g. BlenX [52]; NFsim is based on NRM, making support possible in principle [35]). I will work toward supporting an increasing number of different distributions in ε and the supported underlying simulation algorithms. This makes linking models to observed data easier. Even if all underlying distributions

are exponential, their combined distribution can be non-exponential [31]. Sometimes we can only measure the combined distribution; on such occasions the flexibility to support non-exponential distributions can be important for modeling. Many important "reactions" in systems biology are the combination of large numbers of individual reactions (e.g. translation of mRNA into a protein), making such functionality widely applicable. How to maintain a rigorous mapping (Fig 2) between such arbitrary distributions in stochastic simulations and their expected mean in deterministic ODE solvers is a mathematical problem that will be explored in collaboration with David Anderson at UW-Madison.

Deliverables: We will implement five of the most important distributions of event times for NRM and NFsim (e.g. Gamma, Normal, Lognormal, Constant, Uniform distributions). We will assess the feasibility and work effort needed for mapping the same distributions to deterministic ODE solvers that compute the expected mean.

Goal 2.2. Genetics models with combinatorial explosions

Genetics models are amenable to both deterministic analyses and to stochastic simulations, like systems biology models. However the nature of actions in these systems is fundamentally different. Consider simple viability selection (Fig 6C). As in rule-based reactions, many parts could be affected. On many occasions the timing of selection can be relatively fixed. However, there is more than one possible outcome and this can be modeled stochastically or deterministically. The differences to systems biology become even more explicit in a controlled genetic cross (Fig 6D): the input is well known ('two parents') and the timing of the action is deterministic ("the cross happens"). However, the output of this action is probabilistic; neither the number nor the types produced are known for the many possible offspring genomes (even if expectations are known). The nature of these actions means that simulating genetic systems with traditional systems biology algorithms either requires very cumbersome notation (one action for each potential outcome) or is completely impossible (the resulting uncertainty cannot be represented by the stoichiometry matrices used in systems biology, which determine the changes for each given action). For recombination in discrete generations additional variability exists when it comes to select the parents (Fig 6E).

Deliverables: We will develop and implement algorithms that will allow the rigorous simulation of recombining systems with different combinatorial structures, assuming that the list of possibilities for each combinatorial structure is finite, discrete and easy to specify explicitly. We will collaborate with David Anderson, a mathematician at UW-Madison, to select or develop appropriate algorithms (see letter). We will explore to what degree formalisms described elsewhere (e.g. [53]) might provide opportunities to define elegant language interfaces and summarize our findings.

Goal 2.3. Ecological models integrating with genetics

Considering genetic processes in an ecological context usually means that probabilistic input combines with a probabilistic timing, and leads to a probabilistic outcome. I will explore how to best describe these processes that are completely governed by

probability distributions in the most elegant way possible in order to facilitate their simulation. Processes of this type are needed to describe most organisms realistically.

In addition, I will explore how ε needs to be adapted to facilitate the description of ecological models, including various mechanisms of population density regulation. Systems ecology has a long history [54,55] [56] and has extensively used a wide range of modeling approaches [5,57][58]. A range of modeling tools exists for ecological problems, some more readily available than others (e.g. [59][60][61]). Yet genetics is not easy to include in such models, and distributed computation of different parameter combinations can be complicated. Thus, it would be desirable to support the modeling of ecological processes in ε, to facilitate the integrated analysis of ecological and population genetics problems.

It is clear that substantial progress towards the integration of ecology and genetics will be achieved during this project. However, it is less clear how much work will be needed to fully accomplish this, especially if each type from Fig 6 should be available for each reasonable probability distribution and if automated translation into different analysis types (Fig 2) should be supported as well. My future goals include continuing the integration of genetics and ecology, as well as starting to include spatial aspects (e.g. by including additional solvers such as [62] [63]) and evolution on multiple levels (e.g. endosymbionts [64] or plasmids [65]).

Deliverables: We will develop and implement an algorithm that will allow the rigorous simulation of systems with probabilistic events as input, arbitrary probabilistic rate laws and probabilistic output (Fig 6F).

Aim 3. Build realistic systems biology and ecological genetics models

It is not possible to develop successful software without knowing the problems of its end users. I plan to work closely with experimental biologists who will provide me with the data that I need to build realistic models. I request funds for a graduate student and a postdoc who will implement the key solvers described above and use them to address relevant biological questions in the following systems.

Goal 3.1. Systems biology of VSV viruses

VSV is an excellent model system for understanding non-recombining RNA viruses. The lab of John Yin at UW-Madison had been developing very detailed computational VSV models from the literature, to understand the precise dynamics of the production of various virus components that contribute to fitness [66,67]. Now his lab develops techniques to observe VSV with increasing accuracy [68][69] to build more precise models from direct data. We will compare the efforts needed to build such models by traditional means and by ε. We will then use the ε models to infer various evolutionary parameters [67][70,71], including advantageous mutational effects that had not been inferred before for viruses using this approach [1]. Such parameters are needed for understanding the long-term evolution of VSV in the absence of recombination [72-74].

Deliverables: A VSV virus model based on the data provided by the Yin Lab and predictions of virus growth in cells and advantageous mutational effects in viruses.

Goal 3.2. Copepod ecological genetics

Copepods are small crustaceans that constitute a critical food source for many fisheries throughout the world (e.g. salmon, haddock, pollock), and are of great economic importance [75]. Thus copious amounts of ecological data have been collected for copepods, and a number of ecological models have been built to understand various aspects of their population dynamics (e.g. [75] [75] [76] [77-80]). Copepods also have been proposed as a model organism for ecotoxicology [81]. Understanding the adaptive evolution of copepods is important for predicting their responses to new environmental conditions or to environmental pollution. Yet there are no models that combine realistic ecology with realistic genetics to understand how one impacts the other and how both impact adaptive evolution [82-84]. My goal is to build realistic models of copepod evolution by synthesizing existing ecological data and combining it with genetic data for important adaptive features such as responses to salinity change [85] [83,84]. Recent work collecting genetic and genomic data in copepods offers the potential for constructing increasingly realistic models of evolution [86] [87] [85]. More specifically, we will combine an explicit genetic model of the ion transporters of importance for salinity adaptation [85] with a realistic ecological background model. Such integrated models enhance our ability to determine how copepods adapt to novel environments and allow us to test various hypotheses about the origins of invasive populations [88]. The laboratory of Carol E. Lee at UW-Madison investigates copepod evolution in response to environmental change and would benefit from realistic models of adaptation during copepod invasions into novel habitats. We have formed a collaboration where her lab will provide ecological and genetic data on copepods.

Deliverables: My lab will construct, and analyze a model of adaptive evolution of copepod invasions with a realistic ecological and genetic basis using data provided by the Lee Lab.

Goal 3.3. Additional diverse modeling projects and evolutionary systems biology

My lab is located close to various biological departments where opportunities for collaboration abound. ε needs to prove its usefulness for modeling in many pilot studies. So I want to support undergraduate students interested in building diverse biological models, including ones that build bridges between systems biology and evolutionary biology [1] [70] and models that integrate different disciplines, such as ecology and genetics. The overall goal of my career is to make models of evolution more realistic and link them closer to observational data, making Aim 1+2 pivotal for my career. Workplan:

	Year 1	Year 2	Year 3	Year 4	Year 5
Postdoc	Goal 1.1-1.2 (ε, Condor…)		Goal 3.1 (VSV)	Goal 2.1 (any event distrib)	
PhD student	Goal 2.2-2.3 (Genetics + Ecology)		Goal 3.2 (Copepods)	Goal 2.1	
UnderGrad	Goal 1.3 (HDF5 Standard development)		Goal E.2 (License web app)		
UnderGrad	Goal 3.3 modeling projects using ε, as opportunities arise (teaching etc.)				
Teacher	-			Teacher 1	Teacher 2
PI	ε: design, ANTLR defs; coordination; meetings; papers; outreach,training				

Broader impact and the integration of education and research

My overall vision is to increase interest in quantitative modeling among diverse groups by using exactly the same quality model description language and tools ("ε") for research, education and outreach. This will shift the curve of interest in quantitative modeling (Fig 7), profoundly impacting the future of science. Dissolving the barrier between "teaching-toys" and "real tools" will make it much easier to teach how to build research quality models. This is **transformational** as it enables students, courses and textbooks in biology to use "ε", like "R" is already used in statistics, motivating many to undertake quantitative analyses. To win over people who are scared by modeling-math, I will use analogies and associations with which most people are familiar, such as driving a car: driving a car requires knowledge about how to steer, the rules of the road, but much less maths than building one. Similarly, using ε is easier than implementing ε.

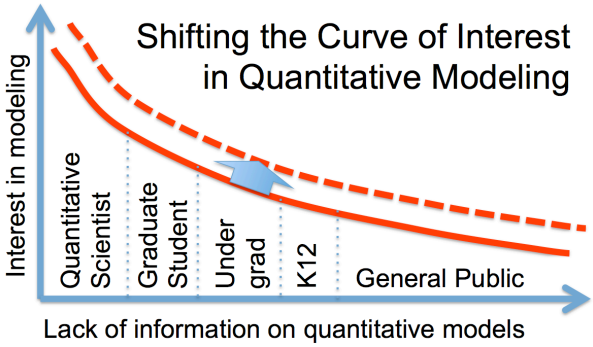

***Fig 7**: Different groups have differing interests in modeling. I aim to use ε to increase interest in quantitative modeling through my broader impact plans.*

Goal E.1. Engaging the interested public: evolution@home participants

My use of publicly contributed CPU-power from evolution@home is an excellent opportunity to engage the public and explain why models of evolution are important. Computing power for projects through evolution@home will be directly proportional to public interest in evolution@home. Such interest is raised by engaging the general public through media work and global computing user communities by contacting multiplicators. I have engaged in media work before (e.g. BBC interview [89]) and attracting >600 CPU years to evolution@home since I started it has taught me what I need to attract even more [38]. People who contribute CPU-power are usually interested in what they are computing. Thus a well-made website can communicate much about science, evolution and modeling. I am revising my website to meet this goal and my group will work to promote the website, the content and the science. We intend to develop games and other engaging tools to draw interest to simulations and increase participation. *Measure of success*: Number of participants, contributed CPU-years.

Goal E.2. 'License for using models': spreading basic modeling knowledge

Appreciating the value of good models comes from understanding a few basic concepts about modeling, which do not require complicated math. I will develop a brief course module explaining the value of good models, the danger of bad ones, how to distinguish if you are not an expert, and how to avoid abusing models. To implement this, I will work with (i) the Delta Program, the local implementation of the NSF-funded Center for the Integration of Research, Teaching and Learning at UW-Madison; (ii) the Crow Institute for the Study of Evolution and the UW Institute for Biology Education; (iii) the two K12

teachers that will join my lab for a summer. Content, name and 'levels of advancement' for this 'license' will be developed in collaboration with education experts and target audiences. Many critical concepts are explained easily by using "dice". To help spread adoption of this 'license', I will offer workshops and work with undergrad software developers to implement the core teaching material as a game-like interactive HTML5 web application that can also be used on iPhones and Android phones. *Measure of Success:* Development of course. Field test and revise with undergraduates and teachers. Web app includes pre- and post- module assessment. Count downloads.

Goal E.3. Inspiring multiplicators: K12 teachers

Teachers are critically important in shaping the modeling skills of the next generation. The Crow Institute for the Study of Evolution and the UW Institute for Biology Education will establish contact with local science teachers. I will invite two to join my lab for the summer to learn about modeling and bring their experience back into class rooms. I will offer this Research Experience for Teachers once ε is more mature (ca. end of year 3), so teachers will have a functioning tool and a few example models that they can take back to school. I request funds for the summer salary of a teacher for 2 years. I expect teachers to help me develop materials that will eventually allow myself and others to effectively reach a K12 audience. We will integrate this with Goal E.2 and test such materials during "Science Saturdays", a regular event drawing many families with children to the Wisconsin Institutes for Discovery, where my lab is located.
Measure of Success: Development of classroom ready materials, workshop materials and outreach activities. Together with the teachers, I plan to offer a teacher workshop at the Wisconsin Society of Science Teachers, where pre- and post- event questionnaires will evaluate changing attitudes. For the materials used at Science Saturdays, we will record how many kids are willing to stay engaged long enough to earn a badge.

Goal E.4. Inspiring underprivileged and minority students to consider modeling

With the right mentoring and training, minority students can become great modelers. I have started mentoring Iratxo Flores-Lorca, a student with Hispanic background. He is making good progress in facilitating the use of the existing 'CVODE' solver in the tools my lab is developing. Later he will also build biological models using his work. I will work with the McNair Program at UW-Madison to recruit first-generation and underrepresented students to engage in research in my lab and, where possible, prepare them for graduate school. Some of these students will be mentored by my graduate students and postdocs, who will have been trained to be effective mentors (see above).
Measure of success: students recruited, change of attitudes towards modeling from pre- and post- course assessments, models built, publications.

Goal E.5. Inspiring future modelers through undergraduate teaching modules

Undergraduate teachers have a large impact on the career choices of the next generation. I hope to inspire the modelers of the future to actually become modelers. I plan to use simplified ε models as examples when teaching undergraduates as part of my 2nd year "*Introduction to Evolution*" lectures and my elective "*Introduction to Evolutionary Systems Biology*" course. This will expose students to the basics of ε and

provide them with a point of contact for more advanced modeling needs they might have later. To develop these materials and the assessment tools to measure their success, I will participate in the Instructional Materials Development course offering through the Delta Program. Through this course, a graduate student or postdoc from my lab and I will develop the learning goals, the teaching materials, and the assessment tools for a model unit. The same student may continue to work with me as a Delta intern to implement the teaching materials in my courses and assess their impact.
Measure of success: Development of model unit. Field testing and revision of unit.

Goal E.6. Training modelers on the job: Postdocs and PhD students

My lab provides an excellent environment for interdisciplinary training, exposing students and postdocs to the many disciplines needed for building good models. ε is an integral part of research in my lab: Biologists use it to build better bio-models and give feedback on ease of use, while more technically oriented researchers work on improving its capabilities. Many students in my lab will use ε to learn about modeling; their feature requests will help developers to advance the tools and mathematicians to devise new algorithms. I plan to participate in Research Mentor Training through the Delta Program at UW-Madison to most effectively mentor these students. I will also encourage grad students and postdocs to participate in Research Mentor Training as they begin to mentor undergraduate students in my lab, thereby training the next generation of effective, interdisciplinary mentors. *Measure of success*: ε-models built, ε bug reports and change requests turnover, publications using ε.

Goal E.7. Engage modeling scientists through collaborations & meetings

Successful integration of ε into research environments will be determined by (i) the quality of the tool and (ii) adoption by fellow scientists. To increase both, I plan to bring together modelers and experimental biologists who are interested in building effective models of particular systems. This will happen at the level of local collaborations (see Aim 3) and scientific meetings that I organize. I will organize local meetings in Madison, in which researchers can explore ε, consider ways to use it for their own research, and provide feedback on how to improve it. To take these discussions to the national and international level, I will apply to appropriate meeting programs, including working groups, workshops, tutorials or short-term visits at the National Institute for Mathematical and Biological Synthesis (NIMBioS) and the National Evolutionary Synthesis Center (NESCent). I will also bring such discussions to the annual meetings on evolutionary systems biology that I started in 2009 ([90]; e.g. with colleagues I will apply for a workshop at the Banff International Research Station). I found such interdisciplinary discussions to be critical for developing innovative approaches that eventually will require tools; hence I want the output of such discussions to help steer future ε work. Meetings, my publications, and open source access to the tools are my core dissemination strategy. *Measure of success*: Participation at ε meetings, feedback from meeting participants about ε, publications, number of code downloads.

Broader impact summary: I will help raise awareness for modeling across a broad range of society, highlighting its importance for responsible decision-making.

1. Loewe L (2009) A framework for evolutionary systems biology. BMC Systems Biology 3: 27.
2. Judson OP (1994) The rise of the individual-based model in ecology. Trends in Ecology & Evolution 9: 9-14.
3. Gillespie DT (2007) Stochastic simulation of chemical kinetics. Annu Rev Phys Chem 58: 35-55.
4. Bart J (1995) Acceptance criteria for using individual-based models to make management decisions. Ecological Applications 5: 411-420.
5. Grimm V, Railsback SF (2005) Individual-based modeling and ecology. Princeton: Princeton University Press.
6. McConnell S (2004) Code complete 2nd Edition. Redmond, Wash.: Microsoft Press. xxxvii, 914 p. p.
7. Regev A, Shapiro E (2002) Cells as computation. Nature 419: 343.
8. Ciocchetta F, Hillston J (2008) Bio-PEPA: a framework for the modelling and analysis of biological systems. School of Informatics, University of Edinburgh EDI-INF-RR-1231: http://www.inf.ed.ac.uk/publications/report/1231.html.
9. Loewe L (2008) Designing a Front-End for Bio-PEPA. In: Gilmore S, editor. Proceedings of the 7th Workshop on Process Algebra and Stochastically Timed Activities, 30-31 July 2008, Edinburgh, UK. http://pastaworkshop.org/proceedings/loewe-pasta2008.pdf.
10. Loewe L, Guerriero ML, Watterson S, Moodie S, Ghazal P, et al. (2011) Translation from the quantified implicit process flow abstraction in SBGN-PD diagrams to Bio-Pepa illustrated on the cholesterol pathway. Transactions on Computational Systems Biology XIII, LNBI 6575: 13-38.
11. Loewe L, Moodie S, Hillston J (2009) Quantifying the implicit process flow abstraction in SBGN-PD diagrams with Bio-PEPA. EPTCS 6: 93-107 http://arxiv.org/abs/0910.1410.
12. Duguid A, Gilmore S, Guerriero M-L, Hillston J, Loewe L. Design and development of software tools for Bio-PEPA. In: Rossetti RMD, Hill RR, Johansson B, Dunkin A, Ingalls RG, editors; 2009; Austin, Texas. IEEE Press. pp. 956-967.
13. Romanel A, Priami C (2010) On the computational power of BlenX. Theoretical Computer Science 411: 542-565.
14. Feret J, Danos V, Krivine J, Harmer R, Fontana W (2009) Internal coarse-graining of molecular systems. Proceedings of the National Academy of Sciences of the United States of America 106: 6453-6458.
15. Calder M, Hillston J (2009) Process algebra modelling styles for biomolecular processes. Transactions on Computational Systems Biology XI: 1–25.
16. Dematté L, Priami C, Romanel A (2008) The BlenX Language: A Tutorial. Lecture Notes in Computer Science 5016: 313–365.
17. Kummer O, Wienberg F, Duvigneau M, Schumacher J, Köhler M, et al. (2004) An Extensible Editor and Simulation Engine for Petri Nets: Renew. Applications and Theory of Petri Nets 2004 : Lecture Notes in Computer Science 3099: 484-493.
18. Smith LP, Bergmann FT, Chandran D, Sauro HM (2009) Antimony: a modular model definition language. Bioinformatics 25: 2452-2454.

19. Faeder JR, Blinov ML, Hlavacek WS (2009) Rule-Based Modeling of Biochemical Systems with BioNetGen. In: Maly IV, editor. Methods in Molecular Biology: Systems Biology. Totowa, NJ: Humana Press. pp. to appear.
20. Danos V, Feret J, Fontana W, Harmer R, Krivine J (2007) Rule-based modelling of cellular signalling. Lecture Notes in Computer Science 4703: 17-41.
21. Calzone L, Fages F, Soliman S (2006) BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge. Bioinformatics 22: 1805-1807.
22. Ramsey S, Orrell D, Bolouri H (2005) Dizzy: stochastic simulation of large-scale genetic regulatory networks. J Bioinf Comp Biol 3: 415-436.
23. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, et al. (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19: 524-531.
24. Parr T (2007) The complete ANTLR reference guide. Lewisville, Tex. Farnham: Pragmatic ; O'Reilly distributor. xx, 361 p. p.
25. Fowler M, Parsons R (2011) Domain-specific languages. Upper Saddle River, NJ: Addison-Wesley. xxviii, 597 p.
26. Loewe L, Moodie S, Hillston J (2009) Technical Report: Defining a textual representation for SBGN Process Diagrams and translating it to Bio-PEPA for quantitative analysis of the MAPK signal transduction cascade. Technical Report, School for Informatics, University of Edinburgh EDI-INF-RR-1334: http://www.inf.ed.ac.uk/publications/report/1334.html
27. Cohen SD, Hindmarsh AC (1996) CVODE, A Stiff/Nonstiff ODE Solver in C. Computers in Physics 10: 138-143.
28. Serban R, Hindmarsh AC (2005) CVODES: the Sensitivity-Enabled ODE Solver in SUNDIALS. Proceedings of IDETC/CIE 2005. Long Beach, CA. pp. LLNL technical report UCRL-JP-200039.
29. Hindmarsh AC, Brown PN, Grant KE, Lee SL, Serban R, et al. (2005) SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers. ACM Transactions on Mathematical Software 31: 363-396. LLNL technical report UCRL-JP-200037.
30. McCollum JM, Peterson GD, Cox CD, Simpson ML, Samatova NF (2006) The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. Computational Biology and Chemistry 30: 39-49.
31. Gibson MA, Bruck J (2000) Efficient exact stochastic simulation of chemical systems with many species and many channels. Journal of Physical Chemistry A 104: 1876-1889.
32. Anderson DF (2007) A modified Next Reaction Method for simulating chemical systems with time dependent propensities and delays. Journal of Chemical Physics 127: 214107.
33. Anderson DF (2008) Incorporating postleap checks in tau-leaping. Journal of Chemical Physics 128: 054103.
34. Anderson DF, Ganguly A, Kurtz TG (2011) Error analysis of tau-leap simulation methods. Annals of Applied Probability accepted: arXiv:0909.4790.

35. Sneddon MW, Faeder JR, Emonet T (2011) Efficient modeling, simulation and coarse-graining of biological complexity with NFsim. Nat Methods 8: 177-183.
36. Loewe L (2002) evolution@home: Experiences with work units that span more than 7 orders of magnitude in computational complexity. In: Bal HE, Löhr K-P, Reinefeld A, editors. Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid2002). Berlin, Germany: IEEE Computer Society. pp. 425-431 (PDF available from http://evolutionary-research.net/).
37. Loewe L (2002) Global computing for bioinformatics. Briefings in Bioinformatics 3: 377-388.
38. Loewe L (2007) Evolution@home: observations on participant choice, work unit variation and low-effort global computing. Software Practice & Experience 37: 1289-1318.
39. Thain D, Tannenbaum T, Livny M (2005) Distributed computing in practice: the Condor experience. Concurrency - Practice and Experience 17: 323-356.
40. HDF5 Group (2011) The HDF5 homepage. http://www.hdfgroup.org/HDF5/.
41. Beaumont MA, Zhang W, Balding DJ (2002) Approximate Bayesian computation in population genetics. Genetics 162: 2025-2035.
42. Nunes MA, Balding DJ (2010) On optimal selection of summary statistics for approximate Bayesian computation. Stat Appl Genet Mol Biol 9: Article34.
43. Toni T, Welch D, Strelkowa N, Ipsen A, Stumpf MP (2009) Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. J R Soc Interface 6: 187-202.
44. BOINC team (2011) Papers and talks on BOINC. http://boinc.berkeley.edu/trac/wiki/BoincPapers.
45. Millard BL, Niepel M, Menden MP, Muhlich JL, Sorger PK (2011) Adaptive informatics for multifactorial and high-content biological data. Nat Methods 8: 487-492.
46. Howison M, Adelmann A, Bethel EW, Gsell A, Oswald B, et al. (2010) H5hut: A High-Performance I/O Library for Particle-Based Simulations. Proceedings of 2010 Workshop on Interfaces and Abstractions for Scientific Data Storage (IASDS10), Heraklion, Crete, Greece http://www-vis.lbl.gov/Research/H5hut/ http://www-vis.lbl.gov/Publications/2010/LBNL-4021E.pdf.
47. Mason CE, Zumbo P, Sanders S, Folk M, Robinson D, et al. (2010) Standardizing the next generation of bioinformatics software development with BioHDF (HDF5). Adv Exp Med Biol 680: 693-700.
48. Group H (2011) The HDF-EOS Earth Observing System Project. http://hdfeos.org/ http://www.hdfgroup.org/hdfeos.html.
49. PyTables (2011) PyTables homepage: Getting the most *out* of your data. http://www.pytables.org/.
50. Hlavacek WS, Faeder JR (2009) The complexity of cell signaling and the need for a new mechanics. Sci Signal 2: pe46.
51. Hlavacek WS, Faeder JR, Blinov ML, Posner RG, Hucka M, et al. (2006) Rules for modeling signal-transduction systems. Sci STKE 2006: re6.

52. Mura I, Prandi D, Priami C, Romanel. A (2009) Exploiting non-Markovian Bio-Processes. Proceedings of 7th Workshop on Quantitative Aspects of Programming Languages (QAPL), ENTCS 253: 83-98.
53. Barton NH, Turelli M (1991) Natural and sexual selection on many loci. Genetics 127: 229-255.
54. Odum HT (1983) Systems ecology : an introduction. New York: Wiley. xv, 644 p. p.
55. Odum HT (1994) Ecological and general systems : an introduction to systems ecology. Niwot, Colo.: University Press of Colorado. xv, 644 p. p.
56. Shugart HH, O'Neill RV (1979) Systems ecology. Stroudsburg, Pa. New York: Dowden, distributed by Academic Press. xiii, 368 p. p.
57. Grimm V, Revilla E, Berger U, Jeltsch F, Mooij WM, et al. (2005) Pattern-oriented modeling of agent-based complex systems: lessons from ecology. Science 310: 987-991.
58. May RM, editor (1976) Theoretical ecology: principles and applications. Oxford: Blackwell Scientific Publications.
59. Meir E (1996) EcoBeaker 1.0 - An ecological simulation program. EcoBeaker Laboratory Guide and the EcoBeaker Program Manual. Sunderland, MA: Sinauer Associates, Inc. 262 p.
60. isee systems (2011) Stella modelling and simulation software. http://www.iseesystems.com/softwares/Education/StellaSoftware.aspx.
61. Gross LJ, et al. (2008) Grid Computing for Ecological Modeling and Spatial Control. http://www.tiem.utk.edu/ITR06/.
62. Glazier J, Swat M, Heiland R, Hmeljak D, Comanescu A, et al. (2011) CompuCell3D: a C++ PDE solver for spatially explicit problems, based on the Cellular Potts Model, with a Python wrapper with an easy to use GUI and the possibility to define model parameters in XML files.: http://www.compucell3d.org/.
63. Hattne J, Fange D, Elf J (2005) Stochastic reaction-diffusion simulation with MesoRD. Bioinformatics 21: 2923-2924.
64. Rispe C, Moran NA (2000) Accumulation of deleterious mutations in endosymbionts: Muller's ratchet with two levels of selection. American Naturalist 156: 425-441.
65. Paulsson J (2002) Multileveled selection on plasmid replication. Genetics 161: 1373-1384.
66. Lim KI, Lang T, Lam V, Yin J (2006) Model-based design of growth-attenuated viruses. PLoS Comput Biol 2: e116.
67. Lim KI, Yin J (2009) Computational fitness landscape for all gene-order permutations of an RNA virus. PLoS Comput Biol 5: e1000283.
68. Duca KA, Lam V, Keren I, Endler EE, Letchworth GJ, et al. (2001) Quantifying viral propagation in vitro: toward a method for characterization of complex phenotypes. Biotechnol Prog 17: 1156-1165.
69. Lam V, Duca KA, Yin J (2005) Arrested spread of vesicular stomatitis virus infections in vitro depends on interferon-mediated antiviral activity. Biotechnol Bioeng 90: 793-804.
70. You L, Yin J (2002) Dependence of epistasis on environment and mutation severity as revealed by in silico mutagenesis of phage t7. Genetics 160: 1273-1281.
71. You L, Suthers PF, Yin J (2002) Effects of Escherichia coli physiology on growth of phage T7 in vivo and in silico. J Bacteriol 184: 1888-1894.

72. Loewe L (2006) Quantifying the genomic decay paradox due to Muller's ratchet in human mitochondrial DNA. Genetical Research 87: 133-159.
73. Loewe L, Lamatsch D (2008) Quantifying the threat of extinction from Muller's ratchet in the Amazon molly (*Poecilia formosa*). BMC Evolutionary Biology 8: 88.
74. Loewe L, Cutter A (2008) On the potential for extinction by Muller's Ratchet in *Caenorhabditis elegans*. BMC Evolutionary Biology 8: 125.
75. Beaugrand G, Edwards M, Legendre L (2010) Marine biodiversity, ecosystem functioning, and carbon cycles. Proc Natl Acad Sci U S A 107: 10120-10124.
76. Ianora A, Miralto A, Poulet SA, Carotenuto Y, Buttino I, et al. (2004) Aldehyde suppression of copepod recruitment in blooms of a ubiquitous planktonic diatom. Nature 429: 403-407.
77. Ceballos S, Ianora A (2003) Different diatoms induce contrasting effects on the reproductive success of the copepod Temora stylifera. Journal of Experimental Marine Biology and Ecology 294: 189-202.
78. Ianora A, Poulet SA, Miralto A (2003) The effects of diatoms on copepod reproduction: a review. Phycologia 42: 351-363.
79. Miralto A, Barone G, Romano G, Poulet SA, Ianora A, et al. (1999) The insidious effect of diatoms on copepod reproduction. Nature 402: 173-176.
80. Miralto A, Guglielmo L, Zagami G, I B, Granata A, et al. (2003) Inhibition of population growth in the copepods Acartia clausi and Calanus helgolandicus during diatom blooms. Marine Ecology-Progress Series 254: 253-268.
81. Raisuddin S, Kwok KW, Leung KM, Schlenk D, Lee JS (2007) The copepod Tigriopus: a promising marine model organism for ecotoxicology and environmental genomics. Aquat Toxicol 83: 161-173.
82. Lee CE, Remfert JL, Chang YM (2007) Response to selection and evolvability of invasive populations. Genetica 129: 179-192.
83. Lee CE, Remfert JL, Gelembiuk GW (2003) Evolution of physiological tolerance and performance during freshwater invasions. Integr Comp Biol 43: 439-449.
84. Lee CE, Petersen CH (2002) Genotype-by-environment interaction for salinity tolerance in the freshwater-invading copepod Eurytemora affinis. Physiol Biochem Zool 75: 335-344.
85. Lee CE, Kiergaard M, Gelembiuk GW, Eads BD, Posavi M (2011) Pumping ions: rapid parallel evolution of ionic regulation following habitat invasions. Evolution in print: doi: 10.1111/j.1558-5646.2011.01308.x.
86. Winkler G, Dodson JJ, Lee CE (2008) Heterogeneity within the native range: population genetic analyses of sympatric invasive and noninvasive clades of the freshwater invading copepod Eurytemora affinis. Mol Ecol 17: 415-430.
87. Stillman JH, Colbourne JK, Lee CE, Patel NH, Phillips MR, et al. (2008) Recent advances in crustacean genomics. Integr Comp Biol 48: 852-868.
88. Lee CE, Gelembiuk GW (2008) Evolutionary origins of invasive populations. Evolutionary Applications 1: 427-448.
89. Loewe L (2008) No sex for all-girl fish species. BBC News UK Scotland: http://news.bbc.co.uk/2/hi/7360770.stm.
90. Loewe L, Papp B, Lercher M, Knight C, Soyer O (2009-2011) Symposia and workshops in Evolutionary Systems Biology. http://evolutionarysystemsbiology.org/meeting/.