A Scalable Multi-scale Framework for Parallel Simulation and Visualization of Microbial Evolution

Vadim Mozhayskiy
Genome Center and
Department of Computer Science
University of California Davis
One Shields Avenue
Davis, CA 95616, USA
+1 (530) 618-2346
mozhaysk@ucdavis.edu

Bob Miller
Department of Computer Science
University of California Davis
One Shields Avenue
Davis, CA 95616, USA
+1 (530) 754-0327
bobmiller@ucdavis.edu

Kwan-Liu Ma
Department of Computer Science
University of California Davis
One Shields Avenue
Davis, CA 95616, USA
+1 (530) 752-6958
ma@cs.ucdavis.edu

Ilias Tagkopoulos
Department of Computer Science
and Genome Center
University of California Davis
One Shields Avenue
Davis, CA 95616, USA
+1 (530) 752-7707
itagkopoulos@ucdavis.edu

ABSTRACT

Bacteria are some of the most ubiquitous, simple and fastest evolving life forms in the planet, yet even in their case, evolution is painstakingly difficult to trace in a laboratory setting. However, evolution of microorganisms in controlled and/or accelerated settings is crucial to advance our understanding on how various behavioral patterns emerge, or to engineer new strains with desired proprieties (e.g. resilient strains for recombinant protein or bio-fuels production). We present a microbial evolution simulator, a tool to study and analyze hypotheses regarding microbial evolution dynamics. The simulator employs multi-scale models and data structures that capture a whole ecology of interactions between the environment, populations, organisms, and their respective gene regulatory and biochemical networks. For each time point, the evolutionary "fossil record" is recorded in each run. This dataset (stored in HDF5 format for scalability) includes all environmental and cellular parameters, cellular (division, death) and evolutionary events (mutations, Horizontal Gene Transfer). This leads to the creation of a coherent dataset that could not have been obtained experimentally. To efficiently analyze it, we have developed a novel visualization tool that projects information in multiple levels (population, phylogeny, networks, and phenotypes). Additionally, we present some of the unique insights in microbial evolution that were possible through simulations in TeraGrid, and we describe further steps to address scalability issues for populations beyond 32,000 cells.

TeraGrid '11, July 18-21, 2011, Salt Lake City, Utah, USA.
Copyright 2011 ACM 978-1-4503-0888-5/11/07...$10.00

Categories and Subject Descriptors

D.1.3 [**Software**] Concurrent Programming – *Distributed programming*; E.1 [**Data**] Data Structures – *Graphs and networks*; I.2.11 [**Artificial Intelligence**] Distributed Artificial Intelligence – *Multiagent systems, Intelligent agents*; I.6.0 [**Simulation And Modeling**] – General; J.3 [**Life and Medical Sciences**] – *Biology and genetics*.

General Terms

Experimentation, Design, Algorithms, Performance.

Keywords

Microbial Evolution, Biological Networks, Simulation, Visualization, Multi-scale Modeling, High Performance Computing.

1. INTRODUCTION: BIOLOGICAL CHALLANGE

All life forms, from microbes to higher vertebrates, are constantly subjected to evolutionary processes that lead to adaptation and phenotypic variation. Whether evolutionary forces lead to new and rapidly evolving species, as in the case of adaptive radiation, or are responsible for phenotypic divergence within a species, the underlying mechanism by which complex behavior arises remains the same: gradual accumulation of selected genetic mutations and epigenetic changes gives rise to a myriad of anatomical, physiological and behavioral expressions. Although the notion that evolution, niche adaptation, and phenotypic variation leads to "endless forms most beautiful" can be traced back to Darwin [1], it was only in the last decades that with the advent of high-throughput sequencing and profiling techniques, we were able to understand the mechanisms by which mutations

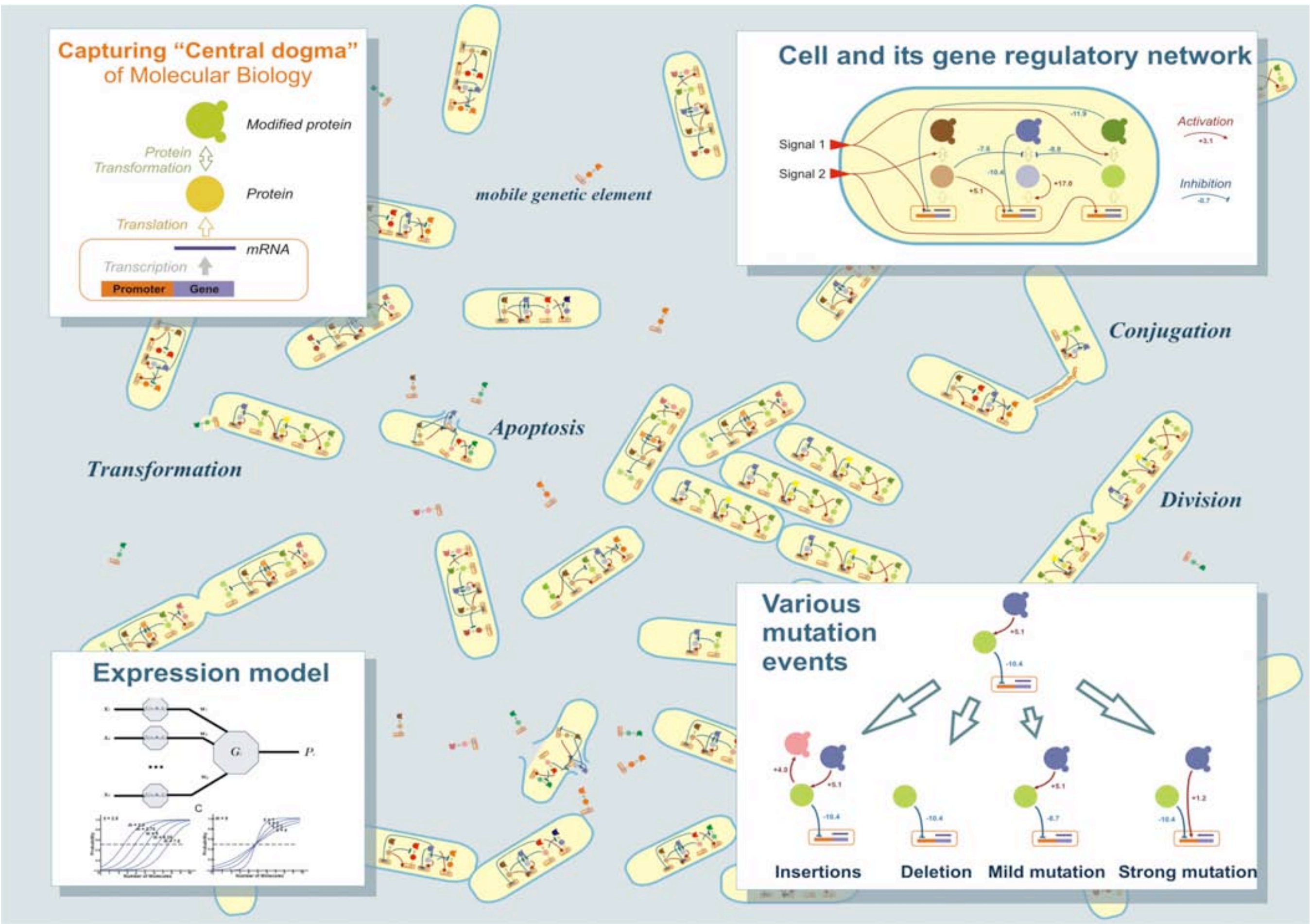


Figure 1. Cartoon in the background shows a model of a cell population. Examples of possible cell events include: division, apoptosis followed by a release of DNA fragments into the environment, and two types of Lateral Gene Transfer: transformation and conjugation. Inserts demonstrate important details of the cell model. From top-left clockwise: (i) “triplet” – mRNA, protein, and modified protein model the central dogma of molecular biology; (ii) gene regulatory network of the cell consists of “triplets”; nodes activate or inhibit each other, the network mutates over time to adopt to the external signals from the environment; (iii) examples of network mutation events: insertion, deletion, mild and strong mutation; (iv) probability of expression of a particular molecule is defined by the sum of regulations by other nodes, each described with a sigmoid function; three parameters are used for each activation or inhibition: regulation strength, midpoint and slope for the sigmoid action function.

give rise to novel traits. Remarkably, it has been shown that even single mutations, such as nucleotide polymorphisms, can yield phenotypes that are significantly dissimilar [2]. The same holds for the rewiring of the gene regulatory and biochemical networks, as they were found to exhibit a high degree of evolvability [3, 4], yet preserve phenotypic robustness when under stabilizing selection and in the presence of disrupting mutations [5, 6].

A challenging task is to identify the environmental and organism-specific characteristics that allow the rapid adaptation from past to new environments. We present a multi-scale simulation framework which helps to investigate various hypotheses in microbial evolution before they can be tested in the laboratory. This includes the microbial evolution simulator and the visualization tool, which will be described in the next sections.

2. BIOLOGICAL MODEL

EVE (Evolution in Variable Environments) simulator employs abstract, multi-scale models of basic sub-cellular phenomena related to expression (transcription, translation, protein modification, degradation, etc.), evolution (mutation, gene duplication, gene deletion, etc.), network regulation and other evolutionary processes such as natural selection (Figure 1). The serial code has been used successfully in the past to generate hypotheses related to regulatory network evolution in nutrient-limited microbial communities [7], and it has been documented elsewhere [8]. The code was recently parallelized to scale to populations of at least 32,000 cells. Further optimization is an ongoing project.

A population is composed of a fixed number of organisms. Each cell is described by its gene regulatory and biochemical network with abstract molecular representations. The network comprises of a number of “triplets” (three nodes): Gene/mRNA, Protein, and Modified Protein (Figure 1, top left panel). The Promoter/Gene/RNA node captures gene regulation and transcription, while the Protein and Modified Protein nodes capture translation and post-translational modification (acetylation, phosphorylation, etc.), respectively. Therefore triplets capture the “central dogma” of molecular biology. Each organism has its own distinct gene regulatory and biochemical network (i.e. a collection of various triplets and weighted regulatory edges) that can be depicted as a directed weighted graph (Figure 1, top right panel).

The probability of molecule creation at each node and at each time step is a function of the regulatory effect of other nodes (activation or inhibitions) on that specific node, and the availability of substrate molecules (Figure 1, bottom left panel). We model the molecule production probability as a two-level sigmoid function that captures a threshold and saturation effects for any given regulator and for the expression of any given node:

$$G_i = basal_i + (1 - basal_i)\cdot \tanh\left(\frac{\sum_{j=1}^{n}\left(w_{ij}\cdot F\left(v_j,\tilde{m}_{ij},\tilde{s}_{ij}\right)\right) - m_i}{s_i}\right)$$

where the sigmoid function F_{ij} describes the regulatory effect of node j on node i:

$$F\left(v_j,\tilde{m}_{ij},\tilde{s}_{ij}\right) = \frac{1}{2}\cdot\left[1+\tanh\left(\frac{v_j-\tilde{m}_{ij}}{\tilde{s}_{ij}}\right)\right]$$

where w_{ij} is the regulatory matrix element (i.e. the strength and direction that exerts node j to node i), v_j is the value of node j, m_i and s_i the midpoint and slope of the target-specific sigmoid function, $\tilde{m}_{ij}$ and $\tilde{s}_{ij}$ the midpoint and slope of the regulator specific sigmoid function, n is number of regulating nodes, $basal_i$ is the basal expression parameter.

In addition to its regulatory network, each organism has a unique metabolic pathway which, when expressed, can metabolize available resources in the environment.

Mutational events (e.g. transcription rate changes, node duplications, node deletions, etc.) occur stochastically at any time point and on any node, thus changing its internal network and potentially its phenotype, which in this context is synonymous to the regulatory and metabolic pathway expression (possible mutational events are shown in Figure 1, bottom right panel). The production and destruction of any molecule has an energy cost, as does the maintenance of molecular species (nodes). Organisms cannot directly sense the presence of resources; however they can potentially infer their future presence, if they are able to process information from various environmental signals through biochemical and regulatory interactions. Once an organism reaches a certain energy level, it undergoes division, increasing its genotype representation in the population, while its progeny replaces an existing organism so that the fixed size of the population is preserved (probability of an organism being replaced is inversely proportional to its energy level in our model).

For our simulations here, we used environments where two signals, s_1 and s_2, carry information regarding the presence of nutrients in the environment (Figure 2). The I/O characteristic of environments A and B is given by the logic *Nutrients Presence [A] = Delayed* (s_1 AND NOT(s_2)) and *Nutrients Presence [B] = Delayed* (NOT(s_1) AND s_2), respectively. This logic produces a single peak when s_1 and s_2 have the temporal characteristics of the waveform presented in Figure 2. Environments that encode an AND, OR and XOR gate were also used. The latter is also the environment with the most complex correlation structure, due to the fact that the XOR gate is not linearly separable. In addition, we introduced a delay in the signal/nutrient correlation to further increase the evolutionary complexity of the environment, as organisms now have to account for it through the topology and dynamics of the respective underlying networks. Similar observations were obtained with the absence of delay, although evolution was faster and resulted in simpler underlying networks.

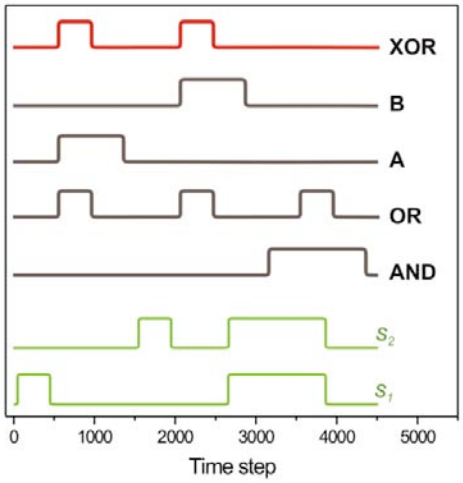


Figure 2. Environments: Environmental signals (green) and nutrient abundance for five environments (bottom to top: AND, OR, A, B, XOR) is a *delayed* function of two signals. One epoch is shown for each environment, which consists of 4,500 time units.

To assess the fitness level of each organism, we report the Pearson correlation between nutrient abundance and response protein expression level over a predefined interval of time, which we call an “epoch” (4,500 time units in our simulations). We stress that this similarity measure is used for visualization purposes as a proxy to each organism’s fitness, and at no point participates or interferes with the selection or evolutionary trajectory of cells during the simulation. High correlation between nutrients and response protein concentration implies an efficient underlying mechanism to metabolize nutrients, as activation of this costly pathway takes place only when it confers an advantage to the organism.

A general structure of the serial version of the code is as follows:

```
Input: population and environment
For (each Epoch)
 For (each Time_step)
  For (each Cell)
    Update_cell()
    if ( Energy == 0 )
      Replace_cell() // new random cell
    if ( Energy > Division_threshold )
      // dividing cell replaces the weakest
      Cell_divide( this-->weakest)
  Output: Phylogenetic information
 Output: Cell fossil history and statistics
```

The model has been extended to incorporate Horizontal Gene Transfer in addition to the other cellular (transcription, translation, modification, growth, death, etc.) and evolutionary (mutation and natural selection) processes. There are three mechanisms for Horizontal Gene Transfer (HGT) by which bacteria can acquire external DNA: transformation, conjugation and transduction (e.g.

review [9]), which we capture through a probabilistic pair-wise model, where an HGT event between any two organisms in the population, or one organism and a genomic “fragment” (e.g. naked DNA present in the solution) occurs with a fixed probability.

In our model a gene and its products are represented by triplets, and therefore HGT can be treated as inter-cellular transfer of one or more triplets. For every HGT event a random subset of triplets (sub-network) is copied from the donor cell and inserted into the regulatory network of the recipient cell. Original regulation of the metabolic pathway RP_0 and triplet T0 by the transferred sub-network is preserved. Upon a parameter sweep for HGT frequency from fully evolved *XOR* networks to non-evolved organisms, we select an “optimal frequency” $5 \cdot 10^{-5}$ (per cell, per time step) of HGT events. It is in the upper range of the experimentally observed values, and consistent with the rest of biological and evolutionary model.

Distribution of the fragment sizes in HGT events may vary greatly in the bacterial world and depends on the type of the transfer and the experimental conditions. However in all three types of HGT the maximum size of the transferred DNA is limited by different parameters: in transduction by the capacity of the viral capsid, in conjugation by the time two organisms stay connected by a pilus, and in transformation by the stability of the naked DNA in the environment. In general the probability of transfering small fragments of meaningful DNA is higher than of larger ones. In our model triplets with preserved regulatory network are transferred from one organism to another, and the fragment size for an established HGT event is chosen using a probability density function as a normalized sigmoid function:

$$P(n) = \frac{1 - \tanh\left(\frac{n-m}{s}\right)}{m \cdot \left(2 + \ln(e^{-\frac{2m}{s}} + 1)\right)} ,$$

where n is the fragment size in triplets, m and s are the middle point and slope of the probability density function, respectively; the denominator is a normalization coefficient. In most cases $s=m$ was used throughout the paper, and therefore 67% of all transferred fragments were not larger than m triplets. The default parameters for HGT were set to $s=m=5$, which results in the expectation value for the size of HGT fragments equal to 4 triplets and slightly smaller than the average size of the minimal network (which is usually 5 to 7 triplets).

To elucidate the *modus operandi* of each evolved network, which may have hundreds of nodes and links, we developed the following heuristic to reduce the network to its “minimal” form, in which only essential nodes and links remain. In this iterative procedure, the fitness effect of a link is assessed after its severance. The link is permanently removed if it is deemed non-essential (less than 5% fitness change). The procedure is repeated until the network cannot be reduced any further. Due to the stochastic nature of the expression model, fitness of a cell can vary as much as 30% between sequential epochs. For that reason the average fitness is evaluated over 10 epochs to reduce that variation to 2%. Multiples iterations over all edges with a tight removal threshold ensure gradual and stable reduction on the network to a near-optimal minimal sub-network.

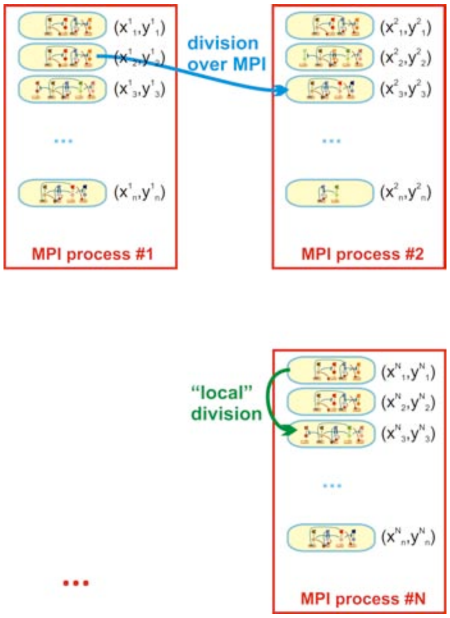


Figure 3. Parallel implementation of simulation framework. Diagram shows data distribution between MPI processes: cell population is divided to run on N MPI processes with n cells per process (total population size n·N).

3. PARALLEL FRAMEWORK

The code is based on a stochastic simulation algorithm where mutational events occur randomly based on predefined probability distributions. We use an MPI model to distribute a population of cells to a set MPI processes (Figure 3). At every time step organisms mutate with predefined probabilities and node values are updated using the stochastic expression model described above; cells which exhausted their energy are removed from the population and replaced with new random cells (to start from a new point on the fitness landscape); cells which reach an energy above the division threshold are duplicated, and

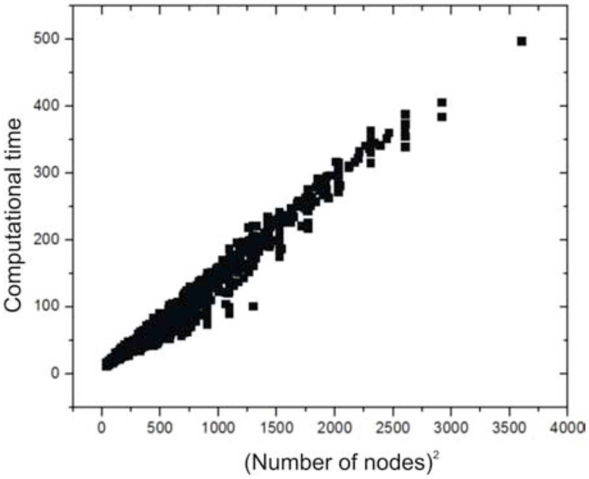


Figure 4 Correlation of the computational time (μsec per organism, per time step) with the square of the number of nodes in its gene regulatory and biochemical network.

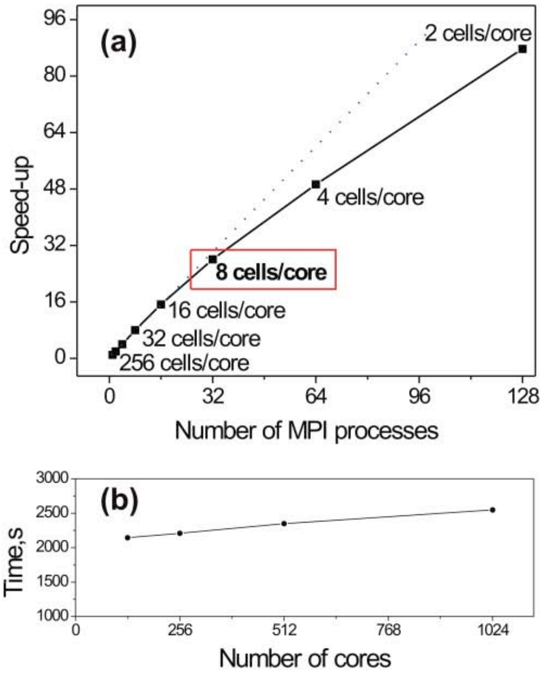


Figure 5. (a) Strong scaling for a population 256 cells. Code scales well for loads of 8 cells/core or more. (b) Weak scaling up to 8192 cells.

daughter cells replace cell with low energies to maintain a constant population size.

The simulations described here are of unprecedented scale and scope, with integrated models of the environment, population, organism, biological network and molecular species. This level of detail is necessary in order to model phenomena that transcend multiple scales, as in the case of Horizontal Gene Transfer. We had to develop efficient algorithms for HPC communication, balancing and process migration, as cell death and division creates unforeseen loads to the various computational cores. In addition, as organisms adapt and evolve, the complexity of their internal networks constantly increases, and with that the need for computational power. Cells with larger networks can be more efficient in nutrients metabolism and therefore grow and divide faster in real time. On the contrary, the computational time for cells with extended genomes is always larger, and scales with $O(K^2)$, where K is the number of nodes within the cellular network (Figure 4). This calls for a synchronization point at each time-point during our simulations, which may lead to poor scalability due to load imbalance.

Initially, cells were distributed to MPI processes with one cell per process per computational core; MPI processes were synchronized at the end of each time step. However, in this initial implementation the imbalance was a problem even for a small number of cells, and the code did not scale beyond 64 cores. The model was improved when a group of cells were assigned to each MPI process, because of averaging effects (i.e. the average computational load was similar among processes). Strong scaling results (Figure 5a) showed that for our problem size, a load of 8 cells per MPI process (per core) was ideal as the imbalance between processes was minimal.

One of the evolutionary trajectories for evolving XOR population is shown in Figure 6. Each step in average and maximum fitness

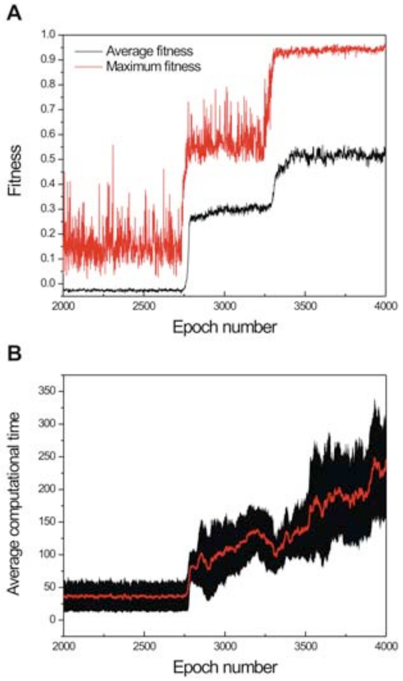


Figure 6. Population evolving in XOR environment. (A) Maximum (red) and average (black) population fitness as a function of time (B) Average computational time (μsec.) per cell per time step (4500 time steps per epoch) is shown in red. Black shade shows the variation (the standard deviation) in the computational time between cells in this evolving population.

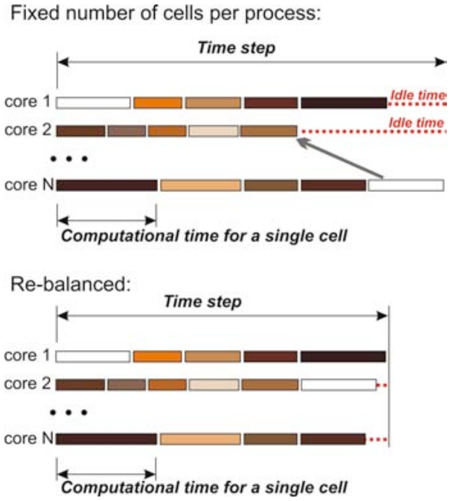

Fixed number of cells per process:
Time step
core 1
core 2
core N
Idle time
Idle time
• • •
Computational time for a single cell
Re-balanced:
Time step
core 1
core 2
• • •
core N
Computational time for a single cell

Figure 7. Dynamic MPI load balancing. Top: in an evolving population, computational time for cells varies with the cell network size. This results in idle cores (dashed lines) with smaller (i.e. fast-to-compute) cells, as they synchronize at each time-point. Each bar segment depicts the computational time of a single cell, and multiple cells, of various complexities, are assigned to a single core. The maximum of these loads (here, the load of core N) defines the speed of the simulation. Bottom: with the addition of a dynamic load balancer, cells are redistributed to minimize idling time.

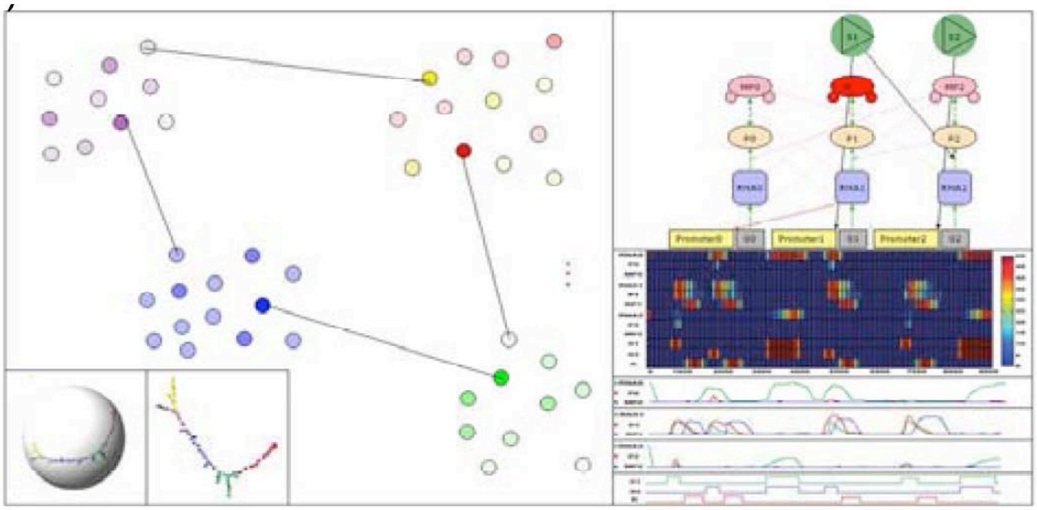


Figure 8. Initial design for multiscale visualization. Upper left panel shows a population clustered according to cell genotype/phenotype (network similarity). In the lower left, the phylogenetic tree is shown divided into clusters based on network similarity. At the upper right, the cellular network diagram for one of the cells is displayed. At the lower right, the expression profiles for the selected cell are displayed both as a heatmap and overlapping line plots.

strongly beneficiary mutation. Averaged computational time for this population is shown in Figure 6B in red. As cells adapt to a XOR environment, gene regulatory and biochemical network of each organism increases, and therefore computational time grows as well. Interestingly, standard deviation of computational time (black shade in Figure 6B) drops after each jump in fitness. Indeed, after one of the cells acquires a strongly beneficial mutation, its descendants quickly outcompete the rest of the population. Resulting population becomes much more uniform, than it was before that mutation, and computational time becomes almost identical for all cells in population. As offsprings accumulate additional mutations, standard deviation in computational time grows until the next fixation of a strongly beneficial mutation. Rare strong beneficial mutations drive evolution; at the same time they are completely unpredictable. This unpredictability could pose a serious challenge for the MPI load balancer, but these events do not increase the load imbalance, and in general almost always decrease it. The rest of the population dynamics is quite continuous and therefore can be predicted well by the load balancer described below.

Next, we further extended our model by implementing a hybrid MPI/OpenMP solution: each MPI process is executed on a multi-core computational node; cells assigned to each MPI process are stored in the node's shared memory; computational cycles for each cell update in an MPI process are dynamically distributed between available cores. Dynamic cell distribution is carried out by creating a pool of cycles and cores, and aims to eliminate the idling time during communication. Weak and strong scaling shows scalability up to 8192 cells with near-linear speedup (Figure 5b).

Finally, by adding dynamic MPI load balancing (Figure 7), computational time is monitored for each cell and is used to redistribute cells between MPI processes in order to have a more balanced load between cores. The cell growth rate is used to predict cell division/death events. Although the current load-balancing implementation is a distributed process, a scalable hierarchical implementation can further increase the performance of the simulator.

We have integrated HDF5 into the EVE simulator as a default format for the “fossil record” of microbial evolution. HDF5 is a library used to store data in an efficient, extensible format that is easily accessible through various APIs. HDF5 uses b-tree indexes to quickly access, allowing for fast random access. The HDF5 format stores the data in a hierarchical, or filesystem-like, structure, as opposed to the relational approach common in most databases. An HDF5 system can span multiple physical files, or exist in a single file. HDF5 is used in EVE to save and load the cell objects. Within each epoch every cell in population at that time point is represented as an HDF5 group.

4. MULTI-SCALE VISUALIZATION

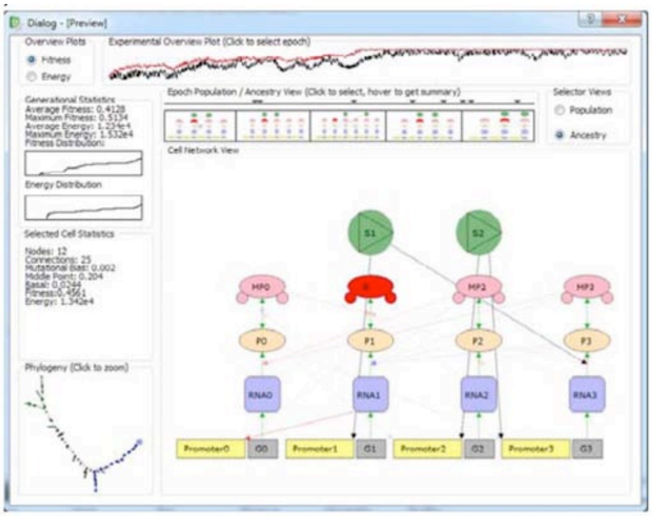
One of the problems encountered during the analysis of the simulation was the enormous amount of data generated, and the lack of existing visualization tools that could handle the relationships between the various types of data produced by the simulator. A multi-scale visualization tool was designed specifically for analysis of simulated evolutionary data.

For effective analysis of the data produced during a simulation run, there are four different scales that need to be considered:

1. **Phylogenetic information:** When tracking how new traits evolve in the population, it is important to be able to determine the relationships between cells at different timesteps. For this purpose, we display a simple node-link diagram of the phylogenetic tree, using the FM3 method for graph layout, as it is shown in the lower left corner of Figure 8 and 9.

2. **Population information:** Some information, such as the development of multiple behavioral groups, requires the analysis of the living population at a moment in time. The left panel of Figure 8 is an example of such a snapshot.

3. **Cellular network information:** In order to determine the mechanism by which a cell process environmental information, it is necessary to view the cell's gene regulatory and biochemical network directly, as in the central panel of Figure 9.



Generational Statistics
Average Fitness: 0.4128
Maximum Fitness: 0.5134
Average Energy: 1.224e4
Maximum Energy: 1.532e4

Selected Cell Statistics
Nodes: 12
Connections: 25
Mutational Bias: 0.002
Middle Point: 0.204
Basal: 0.0244
Fitness:0.4561
Energy: 1.342e4

Figure 9. Multiscale visualization showing phylogenetic tree with ancestors of the selected cell highlighted in blue (lower left), population statistics for the simulation over time (upper left/top), and cellular network diagram for a selected cell (lower right). This view also allows the user to trace the fitness trajectory, zoom in, and trace mutations in underlying networks.

4. **Expression profiles:** Although the cellular network diagram shows how different nodes within the simulated cell influence each other, it provides no insight into the temporal dynamics of expression levels of individual nodes. For this purpose, a microarray-like heatmap or line plot is used as in the lower right panel of Figure 8.

For effective analysis, the visualization tool needs to be able to show any of the above four levels individually, and provide the user with the ability to navigate through the different scales being visualized. From the phylogenetic view, the user is able to either select an individual cell to see its cellular network, or the user can select a group of cells to construct a population snapshot.

Currently, the visualization tool runs on a local machine after the simulation data has been generated, but the next iteration of the tool is intended to run in-situ with the simulation. This would also allow for an on-the-fly visualization to be generated for analysis during the simulation run. This could then be used to control the simulation in progress, to save more data in key portions of the simulation or to filter out unnecessary data to improve simulation speed and to lower storage requirements.

5. APPLICATION: ACCELERATED MICROBIAL EVOLUTION

Recent theoretical predictions [11-12] suggest that evolution generalizes to new environments through facilitated variation, a process in which genetic changes are channeled in useful phenotypic directions. Here we hypothesize that evolution can be accelerated by exposing evolving populations in similar, correlated environments of increasing complexity, and we assess whether Horizontal Gene Transfer (HGT) further accelerates evolution in such settings (these results were published in [12]). When random populations are exposed directly to the XOR environment, more than 4,000 epochs are needed to evolve the delayed *XOR* function (Table 1). In contrast, populations evolve faster in environments of lower complexity, such as the environments A and B. Remarkably, if we sample equal amounts of cells from A and B and expose the new population in the complex environment AB with all other parameters being equal (size of population, average nutrient concentration, etc.), XOR phenotypes of high fitness appear surprisingly fast (Table 1). This effect is even more pronounced in the presence of HGT, where the fittest phenotype arises twice as fast as those without HGT present. Analysis of individual simulation runs results in similar observations, with all experiments leading to phenotypes of increased fitness in the presence of HGT.

Detailed statistics of the evolution probability and speed are shown in Table 1. In "single-step" evolution (un-evolved → XOR) only 18 of 32 (56%) experiments were successful and terminated with an evolved *XOR* population (after 4,000 epochs). Success probability of the "dual-step" adaptation process was estimated as a product of "single-step" probabilities and equals 91% and 82% percent with and without HGT, respectively. HGT accelerates emergence of the combined phenotype in {*A*, *B*} mixed populations by a factor of 1.7. However the probability and the speed of the phenotypic refinement for fitness levels above 0.9 is less affected by HGT relative to the initial emergence of the phenotype above the 0.75 threshold (note that any phenotype with 0.75 Pearson correlation between metabolic pathway expression

Table 1. Rate of adaptation a complex *XOR* environment in different experimental scenarios. The probability and the speed of phenotype emergence are shown for two fitness thresholds 0.75 (evolved organism) and 0.90 (refined evolved organism). Average speed is the average epoch number at which maximum fitness surpasses the threshold.

	Emergence of the organism with fitness *w*			
	w>0.75		*w*>0.90	
	Success Rate	Average speed, *epochs*	Success Rate	Average speed, *epochs*
Un-evolved → *XOR*	18/32	2485	15/32	2489
Un-evolved → *OR*	29/32	1179	13/32	>4,000
OR → *XOR*	30/32	210	5/32	2093
Acceleration by stepwise adaptation	1.8		–	
Un-evolved → *A*	30/32	1043	29/32	1067
Un-evolved → *B*	31/32	1217	31/32	1319
{*A* & *B*} → *XOR*	58/64	234	47/64	448
Acceleration by stepwise adaptation	1.7		1.4	
{*A* & *B*} → *XOR* + **HGT**	64/64	138	48/64	406
Acceleration by HGT	1.7		1.1	

and nutrients exhibits the XOR I/O characteristic). This is to be expected, since subsequent fine-tuning is due to mutations, and not insertion of new functional fragments from other organisms. Evolution through a single environment of intermediate complexity (un-evolved → OR → XOR) accelerates the evolution of a XOR phenotype by a factor 1.8, but with a lower probability of highly fit cells to appear in the final population (only in 5 out of 32 experiments, cells with fitness higher than *w*>0.90 emerged). More details on the effect of HGT on evolution can be found in Reference [12]; the work on guided/accelerated evolution is ongoing and will be published elsewhere.

6. ACKNOWLEDGMENTS

The authors would like to acknowledge Kitrick Sheets (NCSA) for assistance in HPC benchmarking and parallel code development, the members of the Tagkopoulos lab for their comments, and anonymous reviewers for thoughtful suggestions. This work was supported in part by the National Science Foundation through TeraGrid resources (under grants number TG-ASC100043 and TG-ASC110007) and NSF-OCI grant 0941360.

7. REFERENCES

[1] Ciliberti, S., O. C. Martin, and A. Wagner. 2007. Innovation and robustness in complex regulatory gene networks. *Proceedings of the National Academy of Sciences of the United States of America* 104:13591-13596.

[2] Darwin, C. 1859. *On the Origin of Species*. London: John Murray.

[3] Draghi, J., and G. R. Wagner. 2008. Evolution of evolvability in a developmental model. *Evolution* 62:301-315.

[4] Gardner, A., and W. Zuidema. 2003. Is evolvability involved in the origin of modular variation? *Evolution* 57:1448-1450.

[5] Kashtan, N., E. Noor, and U. Alon. 2007. Varying environments can speed up evolution. *Proceedings of the National Academy of Sciences of the United States of America* 104:13711-13716.

[6] Lee, S. H., J. H. J. van der Werf, B. J. Hayes, M. E. Goddard, and P. M. Visscher. 2008. Predicting Unobserved Phenotypes for Complex Traits from Whole-Genome SNP Data. *Plos Genetics* 4:11.

[7] Mozhayskiy, V., and I. Tagkopoulos. 2011. In Silico Evolution of Multi-Scale Microbial Systems in the Presence of Mobile Genetic Elements and Horizontal Gene Transfer, *ISBRA*, vol. accepted, CSU, China.

[8] Parter, M., N. Kashtan, and U. Alon. 2008. Facilitated Variation: How Evolution Learns from Past Environments To Generalize to New Environments. *Plos Computational Biology* 4:e1000206.

[9] Tagkopoulos, I. 2008. Emergence of Predictive Capacity within Microbial Genetic Networks, *PhD Thesis.* Princeton University.

[10] Hachul, S. and M. Junger 2006. An experimental comparison of fast algorithms for drawing general large graphs. *LNCS (Proc. Graph Drawing)* 3843:235-250

[11] Tagkopoulos, I., Y. C. Liu, and S. Tavazoie. 2008. Predictive behavior within microbial genetic networks. *Science* 320:1313-1317.

[12] Thomas, C., and K. Nielsen. 2005. Mechanisms of, and barriers to, horizontal gene transfer between bacteria. *Nat Rev Microbiol* 3:711-21.