Information Technology for Synthetic Biology

Masami Hagiya
Graduates School of Information Science and Technology
University of Tokyo

This short tutorial summarizes the type of information technology that is crucial to ensure progress in synthetic biology. After a brief introduction about the emerging field of synthetic biology, I will explain its importance as an interdisciplinary area between life science and computer science, by showing how life and information can coevolve. I will then explain more about synthetic biology by summarizing current approaches to synthetic biology. Next, I will outline attempts to apply information technology to synthetic biology, including Web-based collaboration, simulation tools, design automation, and laboratory automation. Throughout the tutorial, I describe how both human intelligence and artificial intelligence will likely contribute to the creation of new life through current and next-generation information technology.

Synthetic Biology

The ultimate goal of synthetic biology is to reconstruct life from components obtained using molecular biology and systems biology in a well-defined hierarchical manner. More generally, it is aimed at engineering cells as artificial systems. As explained in a later section (Approaches to Synthetic Biology), various case studies have been conducted to introduce artificial functions into cells by implementing artificial genetic networks. For example, a genetic network consisting of mutually repressing genes, called *repressilator*, was implemented to introduce an oscillator into *Escherichia coli*. Another example is a band detector, a kind of sensor, which recognizes whether the concentration of a specific molecule is within a specified range. Logic gates and memory elements (toggle switches) have also been implemented.

Researchers in the field of nanotechnology are focusing on constructing autonomous molecular systems consisting of sensors, computers, and actuators. DNA molecules are versatile because they can be used for all three kinds of components, and a field referred to as DNA nanotechnology, in which I myself have been involved, has developed over the last few decades. However, living cells have built-in sensors, computers, and actuators with a much wider variety than those of DNA-based components. Engineering living cells for applications that are more suitable for cells should be possible, e.g., bacteria that can search for cancers and destroy them.

However, living cells, even those such as the highly studied *Escherichia coli*, still have unknown components and unexplored mechanisms. Much more research will be required before cells can be freely designed as artificial systems. In contrast, DNA-based systems are relatively simple, and controlling an entire system made of DNA is possible. In this respect, DNA nanotechnology may be considered a warming-up exercise for synthetic biology.

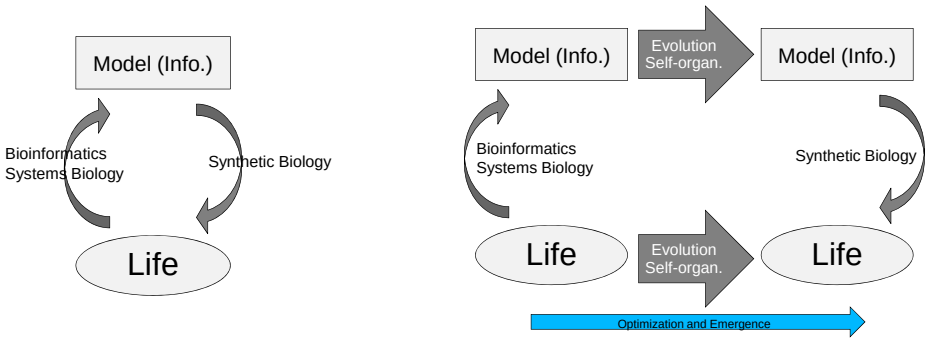
Synthetic biology has a broad range of applications. For example, in addition to developing cancer-killing bacteria, one may be able to develop bacteria that can produce biofuel much more efficiently than natural bacteria, or bacteria that can live in the Martian environment to make the planet habitable. The International Genetically Engineered Machine competition (iGEM) is an annual contest held at MIT, in which teams of undergraduate students from universities all

over the world meet to present their ideas and findings about artificial cells [iGEM2009]. Refer to a later section (Collaboration on a Network) for more detail about iGEM.

Coevolution of Life and Information

Before exploring this issue, one must note that the field of synthetic biology applies models of life (either in the minds of researchers or on computers) and reconstructs life according to such models. Because these models are collections of information, synthetic biology can be said to construct life from information, and can thus be positioned as an interdisciplinary research area between life science and computer science.

Bioinformatics (or computational biology) is also an interdisciplinary area between these two fields. In particular, researchers in one of its subareas, systems biology, consider a cell to be a system and have constructed computer models for this system.



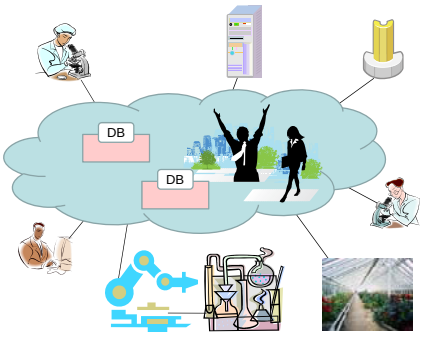
Therefore, bioinformatics and synthetic biology form a kind of loop between life and information (left-hand figure shown above). The implications of this loop are extremely important. Life has evolved for billions of years and adapted to its environment (optimization by evolution). Emergent behaviors in which components self-organize and realize collective functions are also characteristic of life (emergence by self-organization). The loop means that the evolution and self-organization of life can be replaced with similar information-based processes. Furthermore, coevolution of life and information should also be possible.

What do the evolution and self-organization of information entail? This tutorial will provide a simplified view, classifying evolution and self-organization of information into processes conducted using human intelligence and processes conducted using artificial intelligence.

One phenomenon of the former (human intelligence) is the evolution and self-organization of information on the Internet: Wikipedia is a typical example. Each Wikipedia contributor simply submits whatever information he or she chooses, but the total body of knowledge in the system is self-organized and results in a huge but well structured encyclopedia. Wikipedia articles evolve over time in relation with other articles or changes in a certain society or the world; this kind of Web-based activity is referred to as Web2.0. Scientists are also attempting to conduct research in a similar way; this kind of research is being referred to as Science2.0 or Open Science. Synthetic biology was rooted in Science2.0. For example, iGEM is based on the tenets of Science2.0; while iGEM participants are permitted to use genetic components from the standard iGEM registry, they are also required to register their own components made in connection with their iGEM activity. Consequently, the registry becomes richer as iGEM becomes increasingly popular, a process similar to the building of Wikipedia.

In the case of Web2.0, human intelligence is amplified by the network, so knowledge evolves and self-organizes. Another possible way for knowledge to evolve and self-organize is through artificial intelligence. The field of bio-inspired computing has a long history of research, and methods such as evolutionary computation have become a useful tool in many engineering applications for computing optimal parameters or structures for pre-specified purposes. Bio-inspired methods such as evolutionary computation can be applied to the above 'loop' figure by applying methods learned from life to create life.

Combining human intelligence and artificial intelligence may be possible. In a network, programs that behave as humans are called agents. In the future, bio-inspired agents and humans may cooperate in a Web2.0-like evolution and self-organization of knowledge. If devices for conducting experiments related to synthetic biology are connected to a network (in this context, 'cloud' may be a more appropriate term), then research about synthetic biology, including wet-experimental processes, may be conducted entirely on the Web, with human and artificial intelligence cooperating to accelerate the evolution and self-organization of life and information.



A 'coevolution' of life and information could be applied to various situations; e.g., some components may be created by artificial molecular evolution, and whole systems may be designed as models. Automatically designed systems could also be refined through wet evolution. Molecular evolution may also be combined with evolutionary computation.

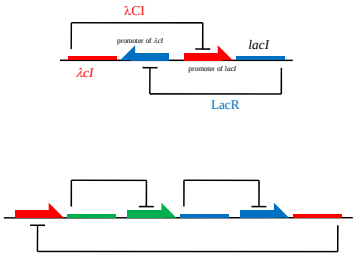
Approaches to Synthetic Biology

Synthetic biology is an emerging field, so researchers are still taking very diverse approaches, which can be broadly classified as described below.

Researchers at MIT, where iGEM is held, take a rather idealistic approach by attempting to introduce engineering disciplines into biology. In particular, they emphasize standardization and abstraction [Endy2005]. (Endy has moved from MIT to Stanford.) More concretely, they are constructing a registry of genetic components (or 'parts'), similar to a transistor data book, and trying to standardize the registry by adding various quantitative data to the component description. Such standardized components can be used without knowledge about their internal implementation. To achieve their goal, they are also applying the strategy of open innovation (called Science2.0 or Open Science), and consider iGEM to be an important step toward their goal.

Another approach is the use of various case studies, which has been the mainstream of synthetic biology. In particular, as mentioned in the first section, researchers have designed and implemented various genetic networks [Sprinzak and Elowitz2005] including a toggle switch [Gardner et al.2000] (upper figure below), an oscillator, called

repressilator [Elowitz and Leibler2000] (lower figure below), a concentration band detector [Basu et al.2005], and logic gates [Ayukawa et al.2007]. Others have tried to introduce unnatural amino acids or unnatural DNA bases into cells [Kiga et al.2002]. Each piece of work is based on a specific biological motivation; e.g., the research about oscillators was conducted to investigate circadian rhythms from the perspective of synthetic biology. Some research focuses on cells as complex systems [Kashiwagi et al.2006].



Distinct lines of research about specific applications have begun to emerge from within the various case studies. For example, researchers are actively investigating the application of synthetic biology to the production of biofuel [Atsumi et al.2008]. As noted above, medical applications in which engineered cells are used as smart drugs are also being explored.

Other researchers are actively investigating life in terms of constructing a so-called minimal cell. They take two main approaches to this goal: one begins with an existing cell and gradually drops non-functioning genes or redundant genes from its genome [Kato and Hashimoto2008]; the other involves constructing a genome from scratch using known genes considered essential to cellular functions. Both approaches require technology for swapping the whole genome of a cell with an artificially synthesized genome [Lartigue et al.2007][Gibson et al.2008][Lartigue et al.2009].

The ‘cell-free’ approach to synthetic biology differs from those mentioned so far. In this line of research, cellular functions are reconstructed from completely well-known components that are placed into a test tube or an artificial cell. *In vitro* protein synthesis is a typical example. Recently, the technology for making and handling a lipid membrane called liposome has matured, enabling construction of cell-like compartments from an artificial membrane. Micro-reactors can also be used as compartments. The field of DNA nanotechnology mentioned above is also likely to begin incorporating this technique.

Steps in Synthetic Biology

This section classifies research activities within synthetic biology into several steps to clarify which kinds of information technology can support the activities at each step, based on the idea of cooperation between human and artificial intelligence described in the previous section.

The research processes involved in synthetic biology can be classified into five main steps.

1. Retrieving existing components
2. Designing a system and identifying new components
3. Simulation of the designed system
4. Implementation of the new components

5. Implementation of the designed system

In the first step, existing components for constructing a target system are retrieved. The standard registry of components, as explained above, will suffice, if all necessary existing components are recorded in the registry. Otherwise, components can be retrieved from various biological databases or the appropriate literature. As noted above, some databases may be constructed in a style similar to that of Wikipedia. This step may also appear during or after the second step.

In the second step, the target system is designed. Some missing components may be identified during this process. Various kinds of network collaboration are possible during this step. Automation is also possible during the system design; e.g., evolutionary computation can be used effectively during design of a genetic network. This will be discussed in more detail later in this tutorial.

In the third step, the designed system is simulated. This step and the previous step may be coupled or iterated. If the simulation of the designed system does not produce an expected result, the system should be redesigned. Parameters of new components may be adjusted by simulation. The field of systems biology, rather than synthetic biology, has developed three methods for simulating cellular activities: ODE, stochastic, and symbolic. Many software tools are already available for these kinds of simulations.

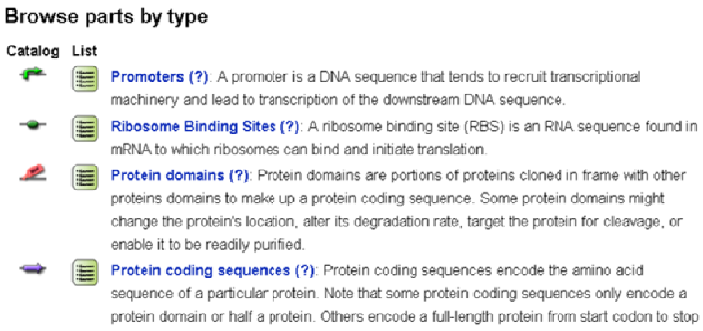
In the fourth step, new components are implemented. In synthetic biology, genes are typical components and are implemented in a plasmid. A set of genes, rather than a single gene, are usually placed into one plasmid. The process of implementing a plasmid may be supported or automated. After experimental results verify an implemented component, it is recorded in the registry along with its quantitative parameters.

In the final step, the whole system is implemented. Again, the process of implementation can be supported or automated.

Based on the above steps, the following sections investigate issues related to network collaboration, simulation tools, design automation, and laboratory automation, exploring how information technology can be used to solve the related challenges and describing what kind of tools are being developed and used.

Collaboration on a Network

As mentioned above, from the inception of iGEM, related activities were structured in a way to encourage Science2.0. The organizers of iGEM are developing a registry of standard biological ‘parts’ using a well-defined hierarchy [parts]. These ‘parts’ are the most basic components, such as protein-coding sequences and ribosome-binding sites (RBS). Devices are constructed using parts and provide basic functions such as protein production. Typical examples of devices include generators that constantly produce useful proteins, and reporters (or measurement devices) that produce GFP (Green Fluorescent Protein) under specific conditions. Systems are formed from devices and can provide some abstract functions. For example, a genetic circuit that functions as a NOT gate is a system.



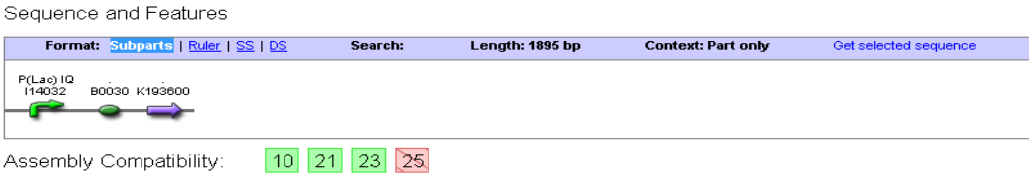
http://partsregistry.org/Main_Page

In 2009, 112 teams from all over the world participated in iGEM. Teams were permitted to use a kit of more than 1,000 parts prepared by the iGEM organizers. These parts were provided in the form of plasmids that were designed for further assembly and distributed on the kit plate. Teams were required to register their own parts and provide plasmids containing those parts. Thus, the registry grows as iGEM becomes increasingly popular.

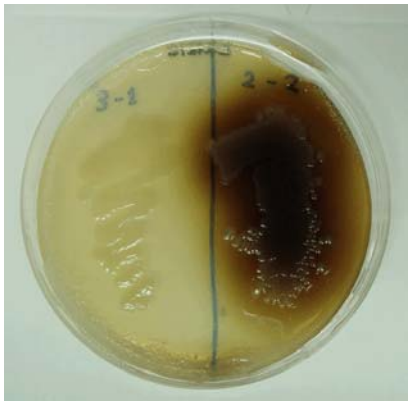
I have collaborated with the Tokyo_Tech team, led by Kiga and Yamamura of the Tokyo Institute of Technology, for years. In 2009, the team's goal was to model the terraforming of Mars using bacteria. The idea was based on science fiction, but the team actually developed four kinds of bacteria that could transform Mars into a habitable planet.

http://2009.igem.org/Team:Tokyo_Tech

They registered 27 parts (and devices), of which 10 were verified as functioning. One part they were particularly proud of was a device that contains the *melA* gene; this produces tyrosinase to blacken *E. coli*, which are intended to control the temperature on Mars.



http://partsregistry.org/cgi/partsdb/pgroup.cgi?pgroup=iGEM2009&group=Tokyo_Tech



http://2009.igem.org/Image:Tokyo_tech_blackecoli3.gif

The team was awarded a gold medal.

The Grand Prix was awarded to the Cambridge team, who developed biosensors that generate four colors (red, orange, brown and violet) for various kinds of inputs. They developed a part called *sensitive tuner*, which is placed in between a sensor and a generator and amplifies the input at an appropriate level.

The registry of biological parts provided by the iGEM organizers is very efficient; iGEM participants report that one can conduct experiments using existing parts only with reference to the registry, i.e., without the need to refer to original sources such as papers and supplementary materials. However, not all parts work as they are intended to. Consequently, parts (and devices) are ranked by their utility; this ranking is currently done individually but automating the process might be possible.

The iGEM registry and other supporting databases have a Wiki-based design. Teams are required to report their findings in a Wiki-based form to allow other teams to refer to their parts, experimental techniques, and ideas. Consequently, these Wiki-based systems serve as an environment for both competition and collaboration.

While Wiki-based systems are good frameworks for collaboration on the Internet, Arita has criticized the use of Wiki systems with regard to their lack of database functionalities for data retrieval and consistency [Arita2009]. He suggested the use of MediaWiki to realize relational databases by Wiki pages [Arita and Suwa2008]. In this system, each page is stored as a row in a relational database, which corresponds to the namespace of the page. This allows more efficient data retrieval and guarantees data consistency by introducing page dependency. Pages that depend on other pages are dynamically generated at the time they are opened using characteristic features of MediaWiki such as *template* and *extension* [Arita and Suwa2008].

A Wiki-page called “OpenWetWare” [OpenWetWare] includes the knowledge required to conduct experiments, in addition to knowledge about biological parts. This page is intended to “promote the sharing of information, know-how, and wisdom among researchers and groups who are working in biology & biological engineering.” Despite such efforts, sharing knowledge about experiments is not easy, and compared to the registry of parts, OpenWetWare is infrequently used during iGEM.

Simulation Tools

Various kinds of simulators have been developed to predict and analyze model behavior. The tools developed for systems biology are also used in synthetic biology and can be classified into three categories.

1. Simulators based on ordinary or partial differential equations
2. Stochastic simulators based on probabilistic models
3. Simulators for symbolic or discrete executions

MATLAB is a typical example of the first category [MATLAB]. E-Cell is the first extensive simulator in systems biology based on differential equations [E-Cell].

With regard to the second category, Gillespie's algorithm [Gillespie1977] and improved tau-leap method [Gillespie2001] is usually used for stochastic simulation [Dizzy]. Some molecular species have very few copies inside a cell, so stochastic simulation is required for precise prediction.

Symbolic or discrete simulations are also used to derive qualitative characterizations of models. Petri nets and process algebra are used for such purposes.

These kinds of simulation methods can be combined. For example, in [Mizunuma and Hagiya2009], we combined a kinetic simulator for ordinary differential equations and Gillespie's algorithm to accelerate the computation required for stochastic simulation. The pioneering work conducted by Matsuno et al. involved the use of hybrid Petri nets, combining methods in the first and third categories [Matsuno et al.2003]. This work was expanded to develop Cell Illustrator, a widely used tool in systems biology that can also be used to visualize biological networks [Cell Illustrator].

For precise and efficient simulation of biological systems, including genetic networks, it is crucial to adopt appropriate models. Design automation, discussed in the next section, also requires suitable models. Research for pursuing good models for synthetic biology should be conducted continuously.

Design Automation

This section focuses on automatic synthesis of artificial biological networks, such as regulatory networks, signaling networks, and metabolic pathways. Even without automation, computer-aided design can assist humans to design such networks, as in other engineering areas. In fact, various tools used in systems biology to model biological networks are readily available for synthetic biology. For example, the Cell Illustrator tool mentioned above was developed to model existing networks but can also be used to design new networks [Cell Illustrator].

Researchers in many areas of science and engineering are working to synthesize models automatically through testing or simulation. In these efforts, models that satisfy a pre-specified condition are generated and examined by a verifier or a simulator. For example, genetic programming is one method used to synthesize LISP programs that satisfy a pre-specified input–output relation by evolutionary computation. For such trials to be successful, they should satisfy the following conditions.

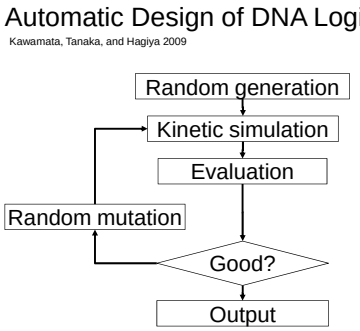
1. Representations of models are intricate.
2. The space of all models is not huge.
3. Verification of a generated model does not take a great deal of time.

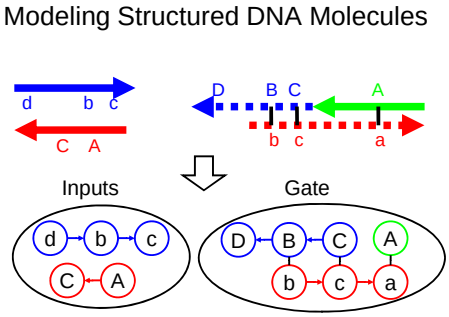
The first condition means that humans cannot easily design new models or even fully understand existing ones. In particular, components are related in complex ways and modification of one component influences the entire model.

Genetic programming does not appear to satisfy this point. Within the domain of programming, humans are far more competent than machines because programming languages are designed for humans. Genetic programming also does not appear to satisfy the second condition; i.e., the space of models was huge. In [Hagiya and Takahashi2000], I surveyed some trials in which systems that satisfy a pre-specified condition were automatically synthesized using a verifier; the attached Appendix briefly outlines these trials and discusses whether they satisfy the above conditions.

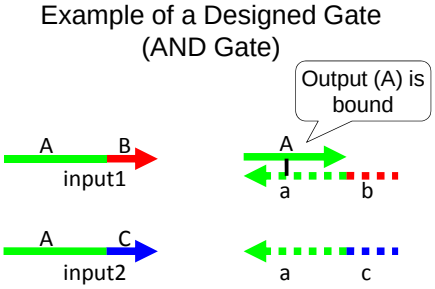
With regard to DNA nanotechnology, we recently tried to synthesize DNA devices such as DNA logic gates and DNA walkers [Kawamata et al.2009]. Behaviors of DNA devices are based on three reactions: hybridization of complementary sequences, its inverse reaction (denaturation of hybridized sequences), and exchange of hybridized sequences (called branch migration). Each reaction is simple, but when their instances are combined, they reveal complex behaviors that cannot be easily predicted by humans. In addition, kinetic simulations based on reaction rates are necessary for correct prediction.

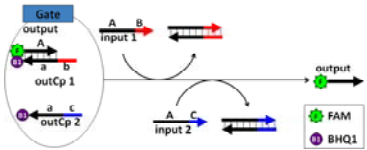
We divided DNA sequences into fragments (units of hybridization) and abstracted DNA devices as graphs whose nodes are fragments (left-hand figure below). The search space for such abstract models is much smaller than the space for concrete models consisting of actual base sequences. Generated abstract models can be kinetically simulated with hypothetical but realizable reaction rates and evaluated according to their behaviors. For example, to search for DNA logic gates, models can be evaluated based on the difference in concentrations (of the output sequence) corresponding to true and false cases. The search for models is performed by simulated annealing based on random generation and partial mutation (left-hand figure below).





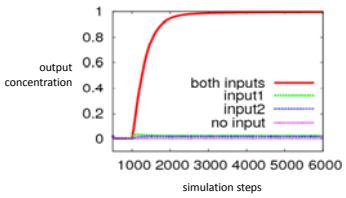
Currently, we are trying to create various DNA devices based on the above framework. To date, we have successfully synthesized DNA logic gates and DNA dials that only respond to signals in a fixed order. In particular, we assigned base sequences to a synthesized AND gate and verified that it actually worked [Kawamata et al.2009].



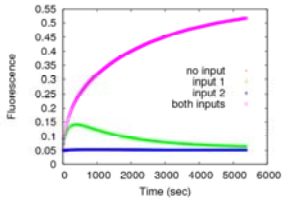


Virtual sequences	DNA sequences
input1	CCAAACTACTTACGTTGAACATACACCGAGGTTT AGTGAACACTTCTAAGCAACTAA
input2	TGAACATACACCGAGGTTTAGTCCAAA
output	TGAACATACACCGAGGTTTAG
outCp1	TGTTCACTAAACCTCGGTGTATGTTCA
outCp2	TTTGGACTAAACCTCGGTGTATGTTCA

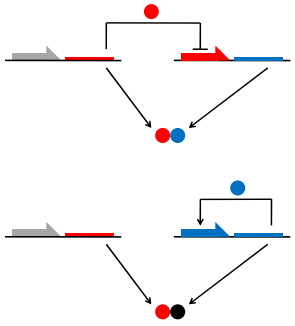
Kinetic Simulation



Experimental Result



Researchers in synthetic biology have conducted similar trials using evolutionary computation. In their pioneering work, François and Hakim were successful in automatically designing bistable switches and oscillating networks [François and Hakim2004]. In their work, genetic networks consisting of genes promoting or repressing each other were randomly generated, mutated, and selected according to the evaluation function corresponding to the function of bistable switches or oscillators. In the case of bistable switches, the concentrations of two proteins A and B were traced in kinetic simulations. At the beginning of the simulation, the concentration of A was high and that of B was low; after a time period T, the concentrations were checked to see if they had been maintained, and then a pulse of B was artificially added to raise the concentration of B. After another period T, the concentrations of A and B were checked to see if they had been reversed. Their model also included dimerization and degradation of proteins. In particular, dimerization of proteins resulted in a new bistable switch with an as-yet unknown structure (figure below). Once a switch or an oscillator was identified, the researchers changed each parameter in the network to determine the parameter range that preserved the network's expected function.



In the case of DNA devices, assigning concrete DNA sequences automatically is possible because thermodynamic parameters for DNA hybridization are available. In contrast, in the case of genetic circuits, implementing automatically designed genetic circuits in which rate constants of various reactions are adjusted during the design process is not a straightforward task. Based on the current technology in molecular dynamics, one cannot design promoters and regulatory proteins completely *in silico*. Another possible direction is to develop a library of regulatory proteins and corresponding promoters with a sufficient variety of parameters, or to automate the experimental process of engineering proteins and promoters.

Laboratory Automation

This final section discusses tools for supporting or automating laboratory experiments. In these steps required for synthetic biology (implementation of the new components and implementation of the designed system), as in the steps required for design, one can both support researchers conducting experiments and ensure that experimental processes are fully automated.

Lab Notebook is one tool used as an online laboratory notebook; it has been adopted as a standard tool in OpenWetWare [Lab Notebook]. It is also based on a Wiki-type system and thus can be shared among collaborators. It can even be made open to the public. Lab Notebook is strongly connected with Science2.0 or Open Science [Science in the open][UsefulChem].

Similar types of commercial software are also available. For example, E-Notebook from CambridgeSoft [E-Notebook] can be used as an online notebook in the fields of chemistry and biology. It can be integrated with various tools such as those used to analyze spectra data and to draw molecular structures.

A successful tool for use as an online notebook should satisfy the following conditions.

1. The burden that a tool imposes on experimenters should be as small as possible. Ideally, such a tool will observe what experimenters do from behind the scenes and automatically record each step of the experiments.
2. The benefit of using a tool should be clear to experimenters. For example, it may be integrated with analysis tools to enable automatic analysis of experimental results, or it may be compatible with presentation software and automatically generate slides or reports.
3. Notebooks can be easily shared with collaborators or made open with well controlled accessibility to guarantee confidentiality and integrity.

Among the benefits offered by an online notebook, traceability of experimental results has become increasingly important, as ethical issues in science have become increasingly serious. Software tools can provide ways to allow researchers to trace experimental results back to raw data and experimental conditions or hypotheses.

One may be able to automate experimental steps in synthetic biology. One example is plasmid construction; during the creation of artificial genetic circuits, plasmids containing genes comprising the target network are constructed through iteration of construction, transformation, overnight culture, and mini-prep. The strategy for constructing the target plasmid is complicated and appears to allow room for optimization or automation. For example, one may be able to ensure automatic scheduling of experimental operations when constructing a target plasmid from existing ones.

Researchers in various fields of science and engineering have investigated optimization of experimental steps executed by robots. For example, Abe et al. applied integer linear programming to optimize operations of a robot that automatically performed experimental operations for DNA computing and related applications [Abe et al.2004]. Evolutionary computation can also be used to optimize or automate experimental processes.

At iGEM 2009, the Best Software Tool was awarded to the Berkeley software team, who integrated the workflow design environment, called Kepler, with the design environment for synthetic biology, called Clotho [Berkeley Software]. The resulting software automatically produces an assembly process represented by a graph, and finally generates commands for a liquid handling robot.

Within the field of systems biology, King et al. constructed a robot that can fully automate the entire process of scientific research, including the generation of hypotheses, planning of experiments, and analyzing experimental results [King et al.2009]. Imagining a robot that could automate research in synthetic biology and create new life would be reasonable.

It would be far more interesting if this robot does not stand alone, but is connected to a network (or 'cloud'). In this situation, all the tools and methods discussed in this tutorial would be applicable together. In particular, human intelligence and artificial intelligence would cooperate and ensure progress within this new research area.

Acknowledgements

I would like to thank Masanori Arita and Daisuke Kiga for their valuable comments on the draft of this tutorial, and Ibuki Kawamata for providing information about iGEM.

References

[Abe et al.2004] S. Abe, M. Hagiya, and T. Nakajima: Code generation for a DNA computer by integer linear programming, *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*, Singapore, 1–3 December, pp. 268–273, 2004.

[Arita2009] M. Arita: A pitfall of Wiki solution for biological databases, *Briefings in Bioinformatics* 10, 295–296, 2009.

[Arita and Suwa2008] M. Arita and K. Suwa: Search extension transforms Wiki into a relational system: a case for flavonoid metabolite database, *BMC BioData Mining* 1 (7), 2008.

[Atsumi et al.2008] S. Atsumi, T. Hanai, and J. C. Liao: Non-fermentative pathways for synthesis of branched-chain higher alcohols as biofuels, *Nature* 451 (7174), 86–89, 2008.

[Ayukawa et al.2007] S. Ayukawa et al.: SYANAC: SYnthetic biological Automaton for Noughts And Crosses, *IET Synthetic Biology* 1 (1–2), 64–67, 2007.

[Basu et al.2005] S. Basu, Y. Gerchman, C. H. Collins, F. H. Arnold, and R. Weiss: A synthetic multicellular system for programmed pattern formation, *Nature* 434, 1130–1134, 2005.

[Berkeley Software] http://2009.igem.org/Team:Berkeley_Software

[Cell Illustrator] http://www.cellillustrator.com/home

[Dizzy] http://magnet.systemsbiology.net/software/Dizzy/

[E-Cell] http://www.e-cell.org/ecell/

[Elowitz and Leibler2000] M. B. Elowitz and S. Leibler: A synthetic oscillatory network of transcriptional regulators, *Nature* 403, 335–338, 2000.

[Endy2005] D. Endy: Foundations for engineering biology, *Nature* 438, 448–453, 2005.

[E-Notebook] http://www.cambridgesoft.com/software/details/?ds=9

[François and Hakim2004] P. François and V. Hakim: Design of genetic networks with specified functions by evolution *in silico*, *Proceedings of the National Academy of Sciences USA* 101 (2), 580–585, 2004.

[Gardner et al.2000] T. S. Gardner, C. R. Cantor, and J. J. Collins: Construction of a genetic toggle switch in *Escherichia coli*, *Nature* 403, 339–342, 2000.

[Gibson et al.2008] D. G. Gibson et al.: Complete chemical synthesis, assembly, and cloning of a *Mycoplasma genitalium* genome, *Science* 319, 1215–1220, 2008.

[Gillespie1977] D. T. Gillespie: Exact stochastic simulation of coupled chemical reactions, *Journal of Physical Chemistry* 81 (25), 2340–2361, 1977.

[Gillespie2001] D. T. Gillespie: Approximate accelerated stochastic simulation of chemically reacting systems, *Journal of Chemical Physics* 115 (4), 1716–1733, 2001.

[Hagiya and Takahashi2000] M. Hagiya and K. Takahashi: Discovery and deduction, *Discovery Science, Third International Conference, DS 2000*, (S. Arikawa and S. Morishita, Eds.), Lecture Notes in Artificial Intelligence, Vol. 1967, pp. 17–37, 2000.

[iGEM2009] http://2009.igem.org/Main_Page

[Kashiwagi et al.2006] A. Kashiwagi, I. Urabe, K. Kaneko, and T. Yomo: Adaptive response of a gene network to environmental changes by attractor selection, *PLOS ONE* 1, e49, 2006.

[Kato and Hashimoto2008] J. Kato and M. Hashimoto: Construction of long chromosomal deletion mutants of Escherichia coli and minimization of the genome, *Methods Mol Biol* 416, 279-93, 2008.

[Kawamata et al.2009] I. Kawamata, F. Tanaka, and M. Hagiya: Automatic design of DNA logic gates based on kinetic simulation, *The 15th International Meeting on DNA Computing and Molecular Programming, DNA15, Preliminary Proceedings*, pp. 8–17, 2009.

[Kiga et al.2002] D. Kiga et al.: An engineered *Escherichia coli* tyrosyl–tRNA synthetase for site-specific incorporation of an unnatural amino acid into proteins in eukaryotic translation and its application in a wheat germ cell-free system, *Proceedings of the National Academy of Sciences USA* 99 (15), 9715–9720, 2002.

[King et al.2009] R. D. King et al.: The automation of science, *Science* 324 (5923), 85–89, 2009.

[Lab Notebook] http://openwetware.org/wiki/Lab_Notebook

[Lartigue et al.2007] C. Lartigue et al.: Genome transplantation in bacteria: changing one species to another, Science 317, 632–638, 2007.

[Lartigue et al.2009] C. Lartigue et al.: Creating Bacterial Strains from Genomes That Have Been Cloned and Engineered in Yeast, *Science* 325, 1693–1696, 2009.

[MATLAB] http://www.mathworks.com/products/matlab/

[Matsuno et al.2003] H. Matsuno, Y. Tanaka, H. Aoshima, A. Doi, M. Matsui, and S. Miyano: Biopathways representation and simulation on hybrid functional petri net, *Silico Biology* 3, 0032, 2003.

[Mizunuma and Hagiya2009] K. Mizunuma and M. Hagiya: Hybrid method for simulating small-number molecular systems, *Algorithmic Bioprocesses* (A. Condon, D. Harel, J. N. Kok, A. Salomaa, and E. Winfree, Eds.), Natural Computing Series, Springer, 2009, pp. 607–622.

[OpenWetWare] http://openwetware.org/wiki/Main_Page

[parts] http://partsregistry.org/Main_Page

[Science in the open] http://blog.openwetware.org/scienceintheopen/

[Sprinzak and Elowitz2005] D. Sprinzak and M. B. Elowitz: Reconstruction of genetic circuits, *Nature* 438, 443–448, 2005.

[UsefulChem] http://usefulchem.wikispaces.com/

Appendix

This appendix reviews the following attempts to conduct and verify automatic syntheses of systems that satisfy a pre-specified condition:

1. Synthesis of machine code (superoptimization)
2. Synthesis of algorithms for parallel garbage collection
3. Synthesis of security protocols

These trials are examined to see whether they satisfied the following three conditions:

1. Representations of models are intricate.
2. The space of all models is not huge.
3. Verification of a generated model does not take much time.

A ‘superoptimizer’ searches for relatively short machine code programs that satisfy a pre-specified condition (or specification). Generated programs are verified by testing using typical inputs or symbolic execution, and synthesized programs should be not only correct but also as short as possible. The possible search space of programs is huge, and applications are limited to peep-hole optimization.

In [Hagiya and Takahashi2000], we tried to synthesize algorithms for parallel garbage collection, where a mutator and a collector work in parallel to manipulate the heap consisting of cells pointing to each other. We enumerated the possibilities of flags in a cell and operations of setting or unsetting each flag, and tested generated algorithms using a verifier (called a ‘model checker’) for a small model consisting of a few cells. We were able to obtain a new algorithm that was previously unknown. The search space was relatively small to allow enumeration of the entire space because each algorithm was obtained through a combination of flags and operations. More importantly, correctness of an obtained algorithm for parallel garbage collection is not intuitively obvious because flag operations interact in complex ways. This also holds true for machines, and model checking takes time.

A security protocol for mutual authentication or key distribution consists of a few messages (two or three in typical protocols) sent and received by participants of the protocol. However, its correctness is not trivial, and many protocols revealed bugs after they were published or deployed. The research group at UCB tried to conduct an automatic synthesis of security protocols using their verifier, which efficiently analyzes symbolic representations protocols. The trial was successful because it satisfied the above three conditions, although it focused on a very limited domain of application.