Genome Engineering

Drew Endy (http://mit.edu/endy/)

Goals Covered Last Time

A. The importance of choosing whether or not a problem is important enough to work on.

B. DNA sequencing & DNA synthesis

Goals for Today

C. Foundational Engineering Concepts: Abstraction, Standards, Insulation, Decoupling

1. DNA synthesis, from genetic information to genetic material. Last class we discussed DNA synthesis technology. Today, gene synthesis companies can provide ~1kb genes for $750 with a two-week or longer turn time. An achievable goal for the next 10 years would be to enable construction of up to 10E6 bp chromosomes for $100,000 with a less than one-week turn time. Automated DNA construction technology of this scale would allow eukaryotic chromosomes to be rebuilt from scratch (e.g., all of yeast, the human Y, et cetera); reverse genetics would become reverse genomics. The impact of automated DNA construction technology on the engineering of biology is similarly straightforward. Engineers learn best through trial and error. Automated construction of DNA would reduce the cycle time and cost of our learning process.

2. Languages and grammars for programming in DNA. Looking ahead, it seems reasonable to anticipate a challenge that will arise as automated DNA construction methods are improved. How will we define what DNA to synthesize? Just as rapid advancements in DNA sequencing capacity through the 1990s forced the biological research community to reactively invest in the development of sequence analysis technologies, advances in DNA construction will require the development of technologies that manage the information that is going into the DNA synthesizers. Here we can adapt and extend two lessons from the history of engineering, abstraction and standardization, which together provide a template for defining the languages and grammars that the biological engineers of the future will depend on as we write many-component genetic programs.

3. Simplicity. The science of biology does a good job of celebrating complexity and exceptions to rules. Practicing engineers often do not enjoy complexity and exceptions to rules. In order to generally realize future biological technologies we need to implement biological simplicity. What does this mean in practice? For example, think about how you program a computer today. The electronics behave as you expect. The software behaves as you expect. Yes, computer software have bugs due to human errors in the programming process but, even though the underlying materials are imperfect, things still behave in a determined fashion. How come? And, importantly, why don't you need to know about all this stuff if you want to program a computer?

4. Foundations for Engineering Biology.
Below from Nature 438, 449-453 (24 November 2005), doi:10.1038/nature04342

"In 1978, Szybalksi and Skalka wrote, "The work on restriction nucleases not only permits us easily to construct recombinant DNA molecules and to analyse individual genes, but also has led us into the new era of 'synthetic biology' where not only existing genes are described and analyzed but also new gene arrangements can be constructed and evaluated." Twenty-seven years later, despite tremendous individual successes in genetic engineering and biotechnology, why is the engineering of useful synthetic biological systems still an expensive, unreliable and ad hoc research process?

The first possibility is that we don't yet know enough about biological systems, or that biological systems are too complex to reliably engineer, or both. For example, some descriptions of natural biological systems are notoriously complex. The large number of unique functional components combined with unexpected interactions among components (for example, pleiotropy) makes it hard to imagine that we might reliably engineer the behaviour of complex biological systems. Furthermore, it is possible that the designs of natural biological systems are not optimized by evolution for the purposes of human understanding and engineering. Thankfully, these concerns are best evaluated by attempting to surmount them.

The second possibility is that the engineering of biology remains a research problem because we have never invented and implemented foundational technologies that would make it an engineering problem. Stated plainly, the engineering of biology remains complex because we have never made it simple (T. F. Knight). As above, the practicality of making biological engineering simple can best be evaluated by attempting to make it simple. Success would help to "create the discipline of synthetic biology: an engineering technology based on living systems". Failures would directly illuminate and help prioritize the most relevant gaps in our current understanding of natural living systems, and suggest how we might best eventually come to understand and apply nature's original technology."

5. How might we make the engineering of biology simple. Most engineering takes place using highly processed and refined materials. How come? Should we consider investing energy in the refinement of the raw materials of life in order to make the engineering of biology simple? If yes then what would we want to do? Well, the history of engineering suggests that there are some basic ideas that are worth considering. Nobody is now sure which of these ideas will be most useful in biological engineering but here's the list of my four favorites: abstraction, standards, insulation, decoupling. What are each of these about? You can read about each of these in more detail via the paper reference in point 4 above. For now here are very quick introductions...

6. Abstraction. Abstraction is an approach for organizing functions and behaviors across a range of levels. The goals is to develop representations for more and more powerful functions, built on top of lower level functions, without being overwhelmed by all the information that is needed to specify the lower level functions. One approach for using abstraction is to develop an abstraction hierarchy, from low to high level functions.

7. Standards. Standards define how things work together (and how things don't work together). Today, there are approximately zero standards that support biological engineering. We would like to have standards that support functional and physical composition of standard biological parts. See the paper from 1864 by William Seller's on screw threads if you need convining. Here's a copy:
http://openwetware.org/images/4/4c/Sellers.1864.pdf

8. Insulation. How do we make parts that behave as expected when we put them together. For example, just because we define a standard for how transcription promoters and open reading frames should be combined doesn't mean that they will naturally follow our arbitrary standard. We might have to implement insulation, so that parts do what we expect them to do but do not interact in unpredictable ways with other parts. Is there insulation inside the cell? Could we make insulation?

9. Decoupling. It's hard to do everything at once. If you have a complicated problem it is oftentimes a very good idea to split the problem into many simpler, individual problems that can be worked on in isolation... The trick is to arrange things so that when you put everything back together the resulting composite object behaves as you desire (see standards and insulation!). For example, think about how the building 32 (Stata Center) was produced. Somebody was (apparently) an expert architect. Somebody else was an expert contractor. And so on. No one person was expert in all aspects of the project. So, one simple form of decoupling in biological engineering will be to separate design of DNA from construction of DNA. This particular decoupling is just starting now.