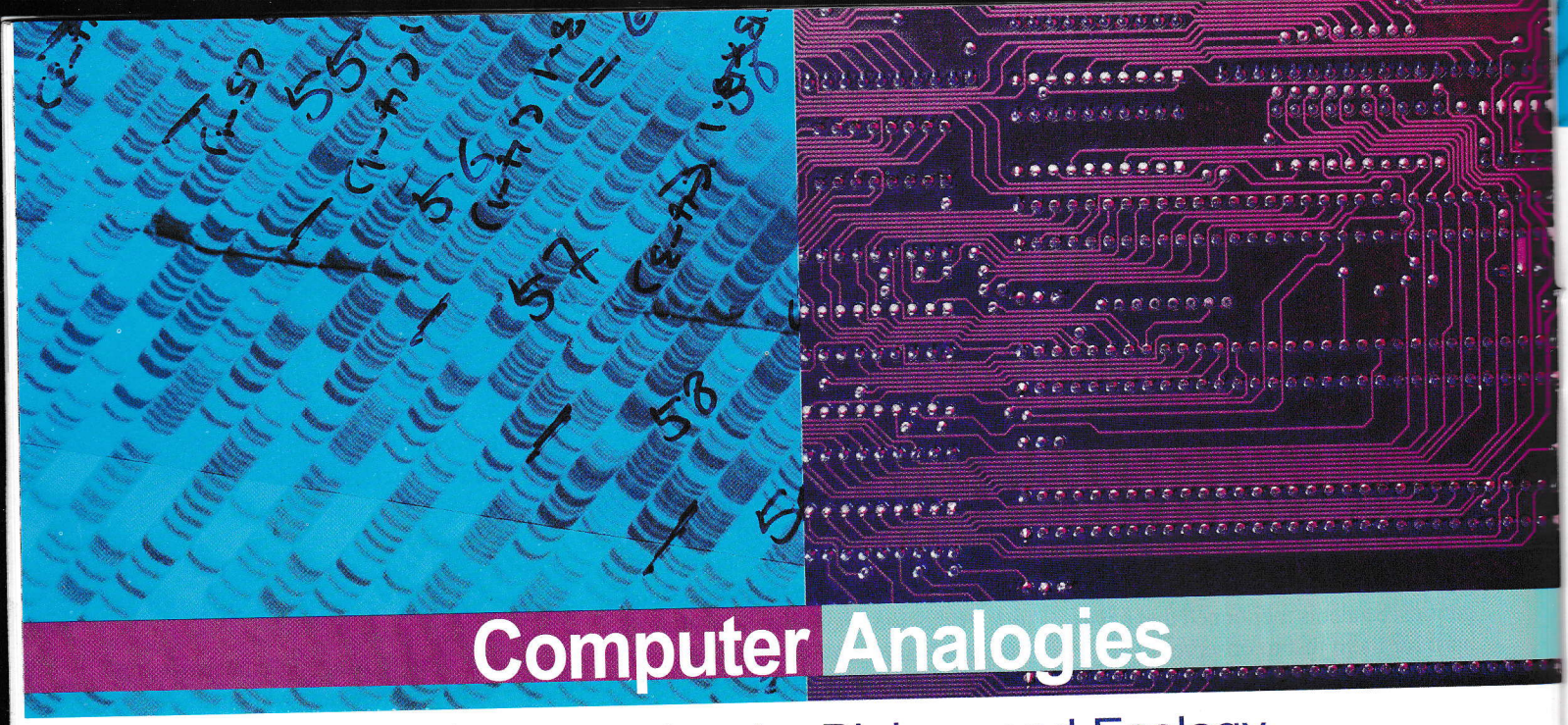


Computer Analogies

Teaching Molecular Biology and Ecology

Stanley Rice and John McArthur

Computer science analogies can aid the understanding of gene expression, including the storage of genetic information on chromosomes and the fate of genes that are no longer used. Computer science analogies can also aid the understanding of species interactions (for example, computer viruses) and natural selection.

Many teachers and authors have used computer science concepts to explain molecular genetics to students or to a general audience (Dawkins 1996a; Coen 1999; Davies 1999). We find that computer science analogies are also useful in explaining ecological concepts. We are not aware of any one publication that brings together computer analogies of molecular and ecological concepts. We do not intend this brief summary as an exhaustive list of analogies between computer science and biology.

We summarize the analogies between biology and computer science listed in Table 1.

Stanley Rice is an assistant professor, department of biological sciences, Southeastern Oklahoma State University, Durant, OK 74701; e-mail: srice@sosu.edu. John McArthur is in the Department of Mathematics and Physics, University of Southern Colorado, 2200 Bonforte Boulevard, Pueblo, CO 81001–4901; e-mail: mcarthur@uscolo.edu.

The alphabet

Computers encode software information in "machine language" with an alphabet of two letters, usually designated 0 and 1. DNA is the software of the cell. It contains the instructions, which must be read and implemented by the hardware, which is pretty much everything else. DNA encodes the cell's software using an alphabet with four letters, now familiar to the public at large: the nitrogenous bases adenine (A), cytosine (C), thymine (T), and guanine (G). Therefore, both cells and computers store information digitally. Four letters can store more information than two, relative to the length of the software message. A byte of computer information, consisting of eight bits, allows 2^8 or 256 options, whereas eight bits of DNA information allow 4^8, or more than 65,000 options. It is therefore conceivable that computers based upon strands of DNA could eventually handle much more complex computations, and much more rapidly, than conventional computers (Winfree and Gifford 2000).

The words

Computers organize bits of information into bytes, each of which specifies a letter or function. Cells organize DNA information into codons, each of which consists of three bases and corresponds to one amino acid.

The files

Computers organize bytes into files, user-defined units of meaningful information. Cells organize codons into genes, each of which specifies the structure of one protein or portion of a protein.

The locations

How can a computer find a file on a disc? The zeros and ones make sense only if the computer knows where to begin reading the sequence. Some computer operating systems use a file allocation table (FAT) or a virtual file allocation table (VFAT) to keep track of where everything is stored on a disc. The FAT maintains an indexed table of locations for all files. A disc has concentric circles called tracks that the computer divides radially into sectors. The outermost circle on a disc is track 0, and its first sector is sector 0. The computer stores the FAT in sector 0 (Figure 1a). If a disc is highly fragmented, the files are in nonadjoining sectors all over the disc

Table 1. *Matrix of biology and computer science concepts presented in this article.*

Biology	Computer science
1. Digital alphabet consists of bases A, C, T, G	1. Digital alphabet consists of 0, 1
2. Codons consist of three bases	2. Computer bits form bytes
3. Genes consist of codons	3. Files consist of bytes
4. Promoters indicate gene locations	4. File-allocation table indicates file locations
5. DNA information is transcribed into hnRNA and processed into mRNA	5. Disc information is transcribed into RAM
6. mRNA information is translated into proteins	6. RAM information is translated onto a screen or paper
7. Genes may be organized into operons or groups with similar promoters	7. Files are organized into folders
8. "Old" genes are not destroyed; their promoters become nonfunctional.	8. "Old" files are not destroyed; references to their location are deleted
9. Entire chromosomes are replicated	9. Entire discs can be copied
10. Genes can diversify into a family of genes through duplication	10. Files can be modified into a family of related files
11. DNA from a donor can be inserted into host chromosomes	11. Digital information can be inserted into files
12. Biological viruses disrupt genetic instructions	12. Computer viruses disrupt software instructions
13. Natural selection modifies the genetic basis of organism design	13. Natural selection procedures modify the software that specifies a machine design
14. A successful genotype in a natural population outcompetes others	14. A successful website attracts more "hits" than others

(Figure 1b); a defragmented disc has the files stored in adjoining sectors (Figure 1c) (Pfaffenberger 1999). The nucleus of the cell does not have a gene-allocation table. Each gene in the cell has a promoter site (in the regulatory region) that indicates where the gene begins. However, just as a file is not necessarily located all in one sector on a disc, so a gene is not necessarily a single uninterrupted stretch of DNA in the cell. A gene can occur at several locations (exons), separated by introns (Alberts et al. 1999) (Figure 2).

Transcription

Both computers and cells transcribe their stored information.

a. For the computer to use the information on the disc, it transcribes this information, still in the form of zeros and ones, into its random access memory (RAM). The file may have existed as separate fragments on a disc, but the computer puts the fragments together in the right order in the RAM transcript. The RAM information is temporary. Furthermore, the RAM contains not only the file information itself but also information about how to use the files (for example, disc operating systems and word processing software) (Pfaffenberger 1999).

b. The cell uses DNA information first by transcribing it into RNA. The genetic information may have existed as separate exons, spaced by introns, but in the finished mRNA, the introns are absent. The computer avoids transcribing junk from a disc because it transcribes only those portions indicated by the FAT. In contrast, the cell transcribes the whole stretch of DNA, starting at the promoter, introns and all, producing heteronuclear RNA (hnRNA). Then small nucleotide ribonucleoproteins snip out the intron transcripts, leaving a finished, or "processed," mRNA molecule (Figure 2). The DNA also stores information that is analogous to word processing software. The cell transcribes this information into transfer RNA (tRNA) and ribosomal RNA (rRNA), both of which become a part of the hardware that puts the genetic information to use (Alberts et al. 1999).The computer information remains on the disc. The computer transcribes, but does not alter, the information on the disc. Therefore the computer can transcribe the file over and over. Similarly, the DNA remains in the nucleus. The nucleus transcribes only the information, not the DNA bases themselves.

Translation

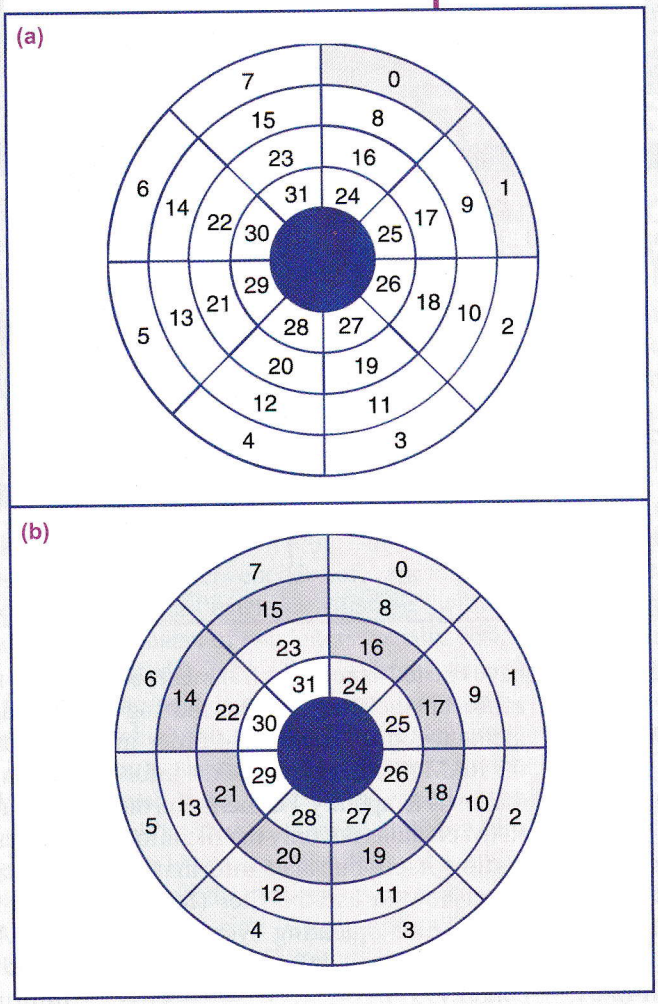
Both computers and cells translate their stored information.

a. In the computer, the RAM contains information that is, like the original file, just zeros and ones. The computer and associated hardware can translate the RAM information into pixels of colored light on a screen or pixels of ink on a paper. This translation process does not wipe out the RAM. Therefore the file can be displayed or printed repeatedly.

b. In the cell, the mRNA directs the translation process. Because each codon determines one amino acid, the sequence of codons in the mRNA determines the sequence of amino acids in the resulting protein. It really is translation, because the information has changed from the nucleic acid language (with its DNA and RNA dialects) to the protein language, which has an alphabet of 20 amino acids and a couple of commands. The necessary hardware includes tRNA and ribosomes (Alberts et al. 1999).

Figure 1. *Organization of computer files on a computer disc. (a) Simplified view of tracks and sectors. Sectors 0 and 1 are the FAT. (b). A fragmented disc.*

SECTORS	CONTAIN
0,1	FAT
2,3,9,10,11,13,22,23,24,25,26	File 1
4,5,6,7,8,27,28	File 2
14,15,16,17,18,19,20,21	File 3
29,30,31	Free



Folders

The computer may organize files into folders, where the user specifies a set of related files. Similarly, the regulatory region of each gene, in addition to identifying the gene's location, also determines the conditions under which the gene will be activated. The nucleus may activate several or many genes simultaneously, or several genes may share a common function, analogous to the files in a folder. Examples include the *lac* operon in the bacterium *Escherichia coli,* and homeobox genes that determine spatial pattern of development (Coen 1999).

Old files

A cell's DNA, like an old floppy disc, contains an invisible record of past activity. The FAT indicates only the files currently in use. When a computer deletes a file, it does not wipe the slate clean, but removes the reference to it in the FAT. The file is still there, but the computer can no longer find it. This is also what happens with genes. Old genes hardly ever go away; they just lose the use of their regulatory regions. These old genes, as well as extra unused copies of genes, have accumulated over time. They contribute to the 99 percent of the DNA in vertebrate nuclei that is not transcribed and is popularly called "junk DNA" (Alberts et al. 1999).

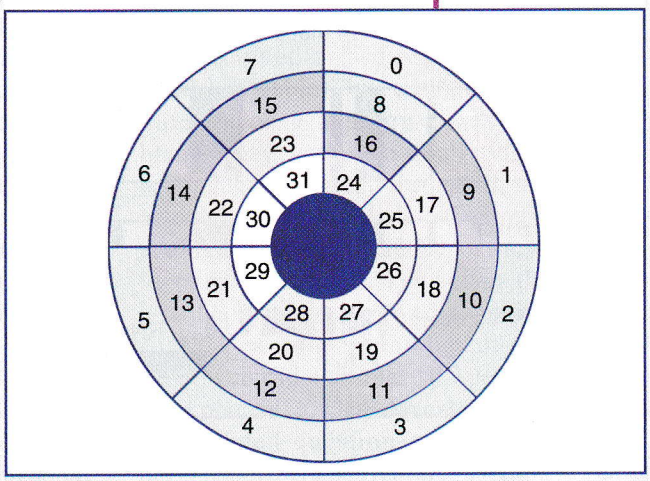
Just as the "junk" files on a disc represent a record of past computer activity, part of the junk DNA is also a record of past genes, and the mutations that have accumulated in them over evolutionary time. Inserting numbers into the FAT can reactivate old files.

Similarly, nuclei can occasionally reactivate old genes. This can sometimes occur spontaneously, as when horses occasionally grow some of the extra toes that their ancestors possessed, but usually it requires artificial stimulation, as in the transplantation of mouse embryo cells that stimulated a chicken embryo to develop teeth (its own teeth, not mouse teeth) (Gould 1983). The bird tooth genes had not been used for millions of years, but were still there, having been copied each generation

The analogy is not perfect. A computer disc is a certain size, and any space not filled with either current or junk files is filled with zeros left over from its original formatting. There is no DNA analog of the truly empty disc space waiting to be filled.

Figure 1(c). *A defragmented disc*

SECTORS	CONTAIN
0,1	FAT
17,18,19,20,21,22,23,24,25,26,27,28	File 1
2,3,4,5,6,7,8	File 2
9,10,11,12,13,14,15,16	File 3
29,30,31	Free



Replication

When a computer copies a disc, it replicates the whole disc, copying all the bits of information. It does not copy one file at a time. This replication process, therefore, copies all of the junk data of discarded files. DNA replication copies all parts of every chromosome, junk and all. In both cases, hardware is needed to do this. In the cell, it is the DNA polymerases and helicase; in the computer, it is the magnetic read/write head.

Diversification

Sometimes you may begin with one original file, then modify it in different ways and give each modified file a new name, producing a family of related files. This is analogous to gene families, in which several genes (such as the globin genes) have been modified for different functions from a single ancestral gene.

Genetic engineering

It is very easy for a computer to insert one file, or information from a notepad, into another file. This is because all the information is in the same digital format. Genetic engineering has been producing transgenic organisms for more than two decades with a high degree of success because all species use the same genetic code. The problem is not in putting the DNA from one species into the chromosomes of another. The problem is, will the resulting mixture work? Will it, when translated, result in an organism whose components function together? (You can mix the parts of a typewriter and a car, but will the result be anything other than a pile of parts?) Our technology is currently crude enough that we are, more or less, shooting genes from one species into the chromosomes of another. This is analogous to inserting a file from one disc into a file on another disc, somewhat at random, which sometimes produces an improved file, but often just creates confusion.

Genetics and Evolution: Darwin in Cyberspace

Computers no longer exist completely in isolation from one another. Now, computers interact, just as individuals interact within populations and ecological communities. We consider two examples.

1. *Computer parasites.* Internet connections allow files to travel from one computer to another, like microbes traveling from one human host to another. Some of these microbes are parasitic, bringing harm to the host. In cyberspace, a computer file traveling from one site to another can be parasitic, harming the recipient like a germ. The analogy between computer viruses and biological sets of viruses is very close because each type of virus is the ultimate parasite—a set of instructions that reproduces the parasite at the expense of the host. Just as different biological viruses infect cells in different ways, computer viruses infect different sectors of software.

A biological virus is a set of nucleic acids, packaged in protein, that instruct your cells about how to make copies of the virus's nucleic acid and its protein. The virus software subverts your cells' replicative hardware. The new copies of the virus can then disperse to new hosts. At the same time, some damage may result to your body. Some viruses operate outside of the cell nucleus, while others insert themselves into the chromosomes. Some viruses are lytic—they cause the cell to die, and have quick and drastic (acute) effects on the host. Others are lysogenic—they blend into the genes and allow a gradual replication of viruses, resulting in a long-lasting (chronic) infection. In contrast, bacteria are cells that reproduce themselves, but cause disease by replicating inside of a host. They do not (with exceptions such as the plant parasite *Agrobacterium tumefaciens*) insert their genes into the host chromosomes.

A computer virus is a set of software instructions that tells your computer to make copies of it and send these copies to other computers. In the meantime the virus instructions may damage your computer software. There are several kinds of computer viruses. File viruses attach themselves to command and executable files (with .com and .exe extensions). When the computer executes the commands in these files, it replicates and spreads the virus. Boot sector viruses are present in the sector of a hard or floppy disc to which the computer refers when booting up when first acti-

vated (Kaspersky Anti-Virus 2000).

Other viruslike software are given biological names. A bacterium is a program that spreads from one system to another by copying itself. It acts independently, rather than infecting other programs. A rabbit is a program designed to exhaust a system by replicating itself without limit. A worm is a program that spreads throughout a network. Finally, computer scientists will sometimes insert software that contains a virus signature into a computer disc in the hope of fooling the computer virus into believing the virus code has already been executed. They refer to this process as immunization!

In biology, the most successful parasites are generally those that do the least amount of damage to the host. The host remains healthy enough to get its food and to circulate among the other members of its species. Natural selection therefore often favors parasites that have the mildest effects on the host and favors hosts that most effectively resist the parasites. The parasitic symbiosis therefore slowly evolves toward commensalism, a process called balanced pathogenicity. It may not work, however, if the parasite disperses very quickly, in which case it does not matter to the parasite that its host has died. By analogy, if a computer virus immediately shut down your computer, that would be the end of the line, and the virus would never travel further. An effective computer virus does not disable your computer until your computer has sent out numerous copies of the virus to other computers. The perfect virus would, of course, be commensalistic and cause no damage whatever, but this would be no fun for the cyberpranksters to write.

2. *Natural selection.* Natural selection is the Darwinian process by which evolution usually occurs. From among the variations that exist in the population, some are more successful at reproduction and therefore are better represented in the next generation (see Weiner 1994 for a nontechnical description). Computers can carry out a similar process, as some software files or subroutines work better than others, and survive into the next iteration or generation. Two examples of natural selection among computer programs follow.

a. *Natural selection in computer-aided design.* Computer simulations of natural selection are common, such as the computer simulation of the evolution of the eye as described in Dawkins (1996b). Computer scientists have also designed evolvable algorithms (Maynard Smith and Szathmáry 1999). Computers, using natural selection algorithms, can design robots (Miller 2000; Pollack 2000).

b. *Natural selection among websites.* Websites are the foci at which browser files on one computer can contact HTML files on another. From this interaction arises evolution. The evolution of the internet seems to follow, to a certain extent, Darwinian principles. Some websites are more successful than others. Their fitness can be measured, as is the fitness of individual organisms, by their reproduction. A successful homepage is one that is replicated many times on the screens of other computers around the world (that is, it has a lot of hits or visits), if the purpose of the website is to convey information. If its purpose is to sell something, its success is measured in terms of the number of sales, not the number of visits. How does natural selection operate in cyberspace?

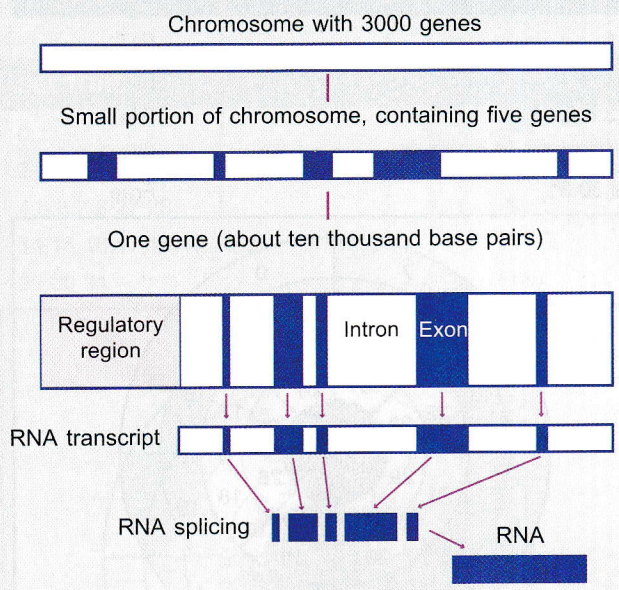
One prerequisite for natural selection is limited resources. It may appear that there is no limit to the number of websites that are possible. However, representation in cyberspace sometimes costs something to the website author; the website creator may conclude that it is not worth the cost if there are few hits or sales. Servers are, moreover, finite in their capacity. There is no more room for every possible website than there is for every offspring of every flower, bird, or mammal.

Another common characteristic of natural selection is competition among the members of a population. Similar websites compete for visitors. Which ones are the most successful?

To answer this question, imagine yourself as a pollinator approaching a field of flowers. What is it that you want from the flower? What is it that attracts you to a certain flower? These are not likely to be the same thing. A flower's bright color attracts you, but what you really want is nectar and/or pollen. The brightest flower is not necessarily the one with the best rewards. Nevertheless, once you visit and explore a flower, you will most likely have pollinated it.

Pollination biology is filled with examples of nonmutualistic interactions between flowers and pollinators or potential pollinators. The hammer

Figure 2. *Storage of genetic information on chromosomes, and transcription. (Adapted from Alberts et al. 1999.)*



orchid, for example, mimics the appearance and scent of a female wasp, attracting male wasps to mate with it. A male wasp must be tricked twice—the first time, he receives the pollen; the second time, he deposits it. Many species of flowers attract flies with their putrid odor, dark meat colors, and location near the ground. These flowers offer the fly no real meat for its maggots. One African water lily even kills some of the flies that visit it. On the other hand, some insects act as nectar thieves by chewing holes in the corolla and drinking the nectar from outside. Clearly, successful pollination depends upon the attraction, not necessarily the reward, of pollinators (see Barth 1991 for more examples).

In cyberspace, the search engine helps you find websites, so the search engine is analogous to the colors and scents of the flowers over which you fly, or surf. Like any pollinator, you have certain things in mind for which you are looking, and you provide this information to the search engine. In the experience of amateur surfers, the search engine rarely finds fewer than 1000 matches, most of which are not helpful. Frequently the surfer unintentionally encounters pornographic page matches. This happens because the search engine reads through the META tags of the web pages. The META tags contain information that will not show up on the screen when the web page is visited. This information can be relevant to the web page, or it can be terms that are almost or wholly irrelevant but which the web page author knows lots of people are looking for. In this way, a web page, like an orchid, can lure visitors that had no intention of visiting it or anything remotely like it. Of course, neither you, nor the pollinator, needs to actually enter. The pollinator can sit on the lip of the corolla and decide to fly away, and you can take one look at the list of matches provided by the search engine and not call up any of the irrelevant ones.

Ten students in a college nonmajors' general biology class at Southeastern Oklahoma State University participated in a discussion on these topics in July 2001. They had previously been introduced to molecular genetics and natural selection concepts in a traditional format. They took a 14-question true-false pre-test, which contained genetics questions but no computer science questions, after this instruction. Then they received a discussion matrix similar to Table 1 but with the computer science analogy boxes left blank. They worked in small groups to suggest analogies 1 through 6, and as a whole class on the remaining analogies. An extra-credit post-test, identical to the pre-test and given the next class period, showed a significant 26 percent improvement (95 percent confidence interval 9 percent to 43 percent, $n = 10$). An extra-credit open-ended question on the next regular quiz asked them to explain a similarity or difference between genetics and computer science. Six students referred to the digital code, one to transcription, and one to viruses, as similarities; two others made less specific references. Similar results were obtained from a general biology class in July 2002.

Using Analogies Wisely

Although we want to avoid giving students the impression that computers or robots are alive, we can still make profitable use of the many similarities between computer science and genetics. Some of these similarities are actual shared concepts, while others are analogies, but both are useful. The instructor can adjust the level of difficulty to suit either general education or advanced biology and computer science classes. We hope that this will improve both biology and computer science education, and encourage the awareness of the overlap between what are often seen as totally distinct fields of interest.

Acknowledgment

The authors would like to thank A. Lisette Rice for helpful background research for this manuscript.

References

Alberts, B. et al. 1999. *Molecular Biology of the Cell,* 3rd ed. New York: Garland.

Barth, F. 1991. *Insects and Flowers: The Biology of a Partnership.* Trans. M.A. Biedeman-Thorson. Princeton: Princeton University Press.

Coen, E. 1999. *The Art of Genes: How Organisms Make Themselves.* New York: Oxford University Press.

Davies, P. 1999. *The Fifth Miracle: The Search for the Origin and Meaning of Life.* New York: Simon and Schuster.

Dawkins, R. 1996a. *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe Without Design.* Reissued. New York: Norton.

Dawkins, R. 1996b. *River Out of Eden: A Darwinian View of Life.* New York: BasicBooks.

Gould, S.J. 1983. "Hen's Teeth and Horse's Toes." In *Hen's Teeth and Horse's Toes* (pp. 177–186). New York: Norton.

Kaspersky Anti-Virus 2000. Computer virus classification. Available online at *www.metro.ch/avpve.*

Maynard Smith, J., and E. Szathmáry. 1999. *The Origins of Life: From the Birth of Life to the Origins of Language.* New York: Oxford University Press.

Miller, J., ed. 2000. *Evolvable Systems: From Biology to Hardware.* Lecture Notes in Computer Science, volume 1801. Berlin: Springer.

Pfaffenberger, B. 1999. *Computers in Your Future.* Upper Saddle River, N.J.: Prentice Hall.

Pollack, J.B. 2000. Dynamical and Evolutionary Machine Organization. Available online at *www.demo.cs.brandeis.edu.*

Weiner, J. 1994. *The Beak of the Finch: A Story of Evolution in Our Time.* New York: Knopf.

Winfree, E., and D.K. Gifford, Eds. 2000. *DNA-Based Computers V.* Providence, R.I.: American Mathematical Society.