Sequence Logos, Machine/Channel Capacity, Maxwell's Demon, and Molecular Computers: a Review of the Theory of Molecular Machines

Thomas D. Schneider*

version = 1.72 of nano2.tex 2000 Oct 21

Reference:

T. D. Schneider. Sequence logos, machine/channel capacity, Maxwell's demon, and molecular computers: a review of the theory of molecular machines. *Nanotechnology*, 5(1):1–18, 1994.

*National Cancer Institute, Frederick Cancer Research and Development Center, Laboratory of Experimental and Computational Biology, National Cancer Institute, P. O. Box B, Building 144, Room 469, Frederick, MD 21702. email address: toms@ncifcrf.gov.

Abstract

Living cells contain many molecules which can make simple decisions, such as whether to bind to a particular nucleotide sequence or not. A theory describing the practical limits of these molecular machines is reviewed. Level 0 theory uses Shannon's information theory to study genetic control systems. Level 1 theory uses Shannon's channel capacity theorem to explain how these biological molecules are able to make their decisions precisely in the face of the thermal maelstrom surrounding them. Level 2 theory shows how the Second Law of Thermodynamics defines the exact extent of the choices available to a molecular machine when it dissipates a given amount of energy. Even the famous Maxwell demon must obey this result. The theory also has implications for designing molecular computers.

1. Introduction

The idea of nanotechnology is founded on the premise that it will be possible to construct machines with atomic scale precision (Feynman, 1961; Drexler, 1981; Drexler, 1986). Biology provides many examples that this is possible; we "merely" need to learn what has been achieved by evolution and to copy it. But eventually we must determine what the engineering limitations of molecular machines are. What should we attempt and what would be foolish to try to violate? This paper reviews a general theory of molecular machines which begins to address this question. The theory is divided into several hierarchal levels.

First, to get a good grasp on these problems, we need practical examples which we can play with in the lab. Because genetic control systems read linear sequences of nucleotides, they provide us with easy access to their coded structure. Level 0 molecular machine theory shows how we can use these systems to study information processes at the molecular level. We will show how we have used information theory (Pierce, 1980; Shannon, 1948; Shannon & Weaver, 1949; Sloane & Wyner, 1993) to dissect the operation of molecular machines (Schneider *et al.*, 1986; Schneider, 1988; Schneider & Stormo, 1989; Schneider & Stephens, 1990; Herman & Schneider, 1992; Stephens & Schneider, 1992; Papp *et al.*, 1993). Other groups are now using these techniques to study a variety of molecular systems (Berg & von Hippel, 1987; Mars & Beaud, 1987; Berg & von Hippel, 1988; Eiglmeier *et al.*, 1989; Stormo, 1990; Fields, 1990; Penotti, 1990; Penotti, 1991; Shenkin *et al.*, 1991; Day, 1992).

Level 1 theory (Schneider, 1991*a*) explains the amazingly precise actions taken by these molecules. For example, the restriction enzyme *Eco*RI scans across double helical DNA (the genetic material) and cuts almost exclusively at the pattern 5′ GAATTC 3′, while avoiding the $4^6 - 1 = 4095$ other 6 base pair long sequences (Polisky *et al.*, 1975; Woodhead *et al.*, 1981; Pingoud, 1985). How *Eco*RI is able to do this has been somewhat of a mystery because conventional chemical explanations have failed (Rosenberg *et al.*, 1987). According to level 1 theory, molecular machines such as *Eco*RI are constrained in their operations by their "machine capacity", which is closely related to Claude Shannon's famous "channel capacity" (Shannon, 1949). So long as one does not exceed the channel capacity, Shannon's theorem guarantees that one may have as few errors in communication as desired. It is this theorem which has led to spectacularly clear telephone communications and compact disk (CD) sound recordings. The equivalent statement for molecular machines is that, so long as a molecular machine does not exceed its machine capacity, it may take actions as precise as may be required for evolutionary survival. We will sketch the proof of this amazing result.

Level 2 theory (Schneider, 1991*b*) deals with the ancient problem of Maxwell's demon (Leff & Rex, 1990) and shows that there is an energetic cost to molecular operations: at least $\mathcal{E}_{min} = k_{\mathrm{B}}T \ln(2)$ joules must be dissipated to the surroundings for every bit of information gained by the machine (where k_{B} is Boltzmann's constant and T is the temperature in kelvin). In the recent literature some authors often claim that $\mathcal{E}_{min}$ is not a limit at all or that information *loss* (rather than gain) is associated with dissipation (Landauer, 1991). However, this relationship is merely (!) a restatement of the Second Law of Thermodynamics

(Schneider, 1991*b*), so those who contest it are unlikely to be correct. We will investigate the source of this confusion.

Finally, we discuss the implications of these results for the prospects of constructing our own molecular machines and molecular computers (Bradley, 1993).

The purpose of this review is to provide a meeting ground for biologists, physicists, computer scientists and electrical engineers. Unfortunately ideas that are famous in one field are unknown in another. Sadly, modern science is now so fragmented that few people know both what a bit is and what translation is. Rather than turn this review into a book, I encourage the reader to read Pierce (1980) for background on information theory, and Watson (1987) or Gonick (1983b) for excellent introductions to molecular biology. Only the central ideas and results of the theory of molecular machines are reviewed in this paper, so only some mathematical equations are presented. Detailed derivations may be found in the references.

2. Level 0 Molecular Machine Theory: Nucleic Acid Binding Molecules as Practical Examples

A ribosome is a collection of proteins and RNAs which reads messenger RNA and uses that information to construct proteins. This translation process starts in a region called the ribosome binding site (Gold *et al.*, 1981; Stormo *et al.*, 1982*b*). One problem facing ribosomes is to locate the binding sites. The cell's survival depends on how well this is done. Some genes are crucial because the translated protein is required for an important cellular function. Other proteins are needed for efficiency, and so the loss of them would put the organism at a competitive disadvantage. If a protein were unnecessary, mutations in its gene would eventually destroy it, and the ribosome binding site at the start of the gene would atrophy. Likewise, if the ribosome were to start translation in places that it shouldn't, the cell would waste energy making useless proteins. Thus it would make biological sense if the only places ribosome binding sites exist is in front of functional genes. However, the conventional wisdom of biologists and chemists says that this is not possible because chemical reactions can have many side products, and only a portion of the substrate becomes product. Thus in the chemists way of thinking, the ribosome can begin translation just about anywhere, with a smooth gradation between real sites and other places. In contrast, information theory says that precise choices (distinguishing sites from non-sites) can be made by the appropriate combination of many "sloppy" components. Whether such precise recognition is actually done is a question which can only be answered experimentally.

The bacterium *Escherichia coli* has approximately 2600 genes (Schneider *et al.*, 1986), each of which starts with a ribosome binding site. These have to be located from about 4.7 million bases of RNA which the cell can produce (Kohara *et al.*, 1987). So the problem is to locate 2600 things from a set of 4.7×10^6 possibilities, and not make any mistakes. How many choices must be made?

The solution to this question, $\log_2(4.7 \times 10^6/2600)$ bits, is "obvious" to those of us versed in information theory, but the reasoning behind how this works in a biological system is subtle and not obvious, so let's consider a simpler example (Fig. 1). Suppose I have 8 boxes

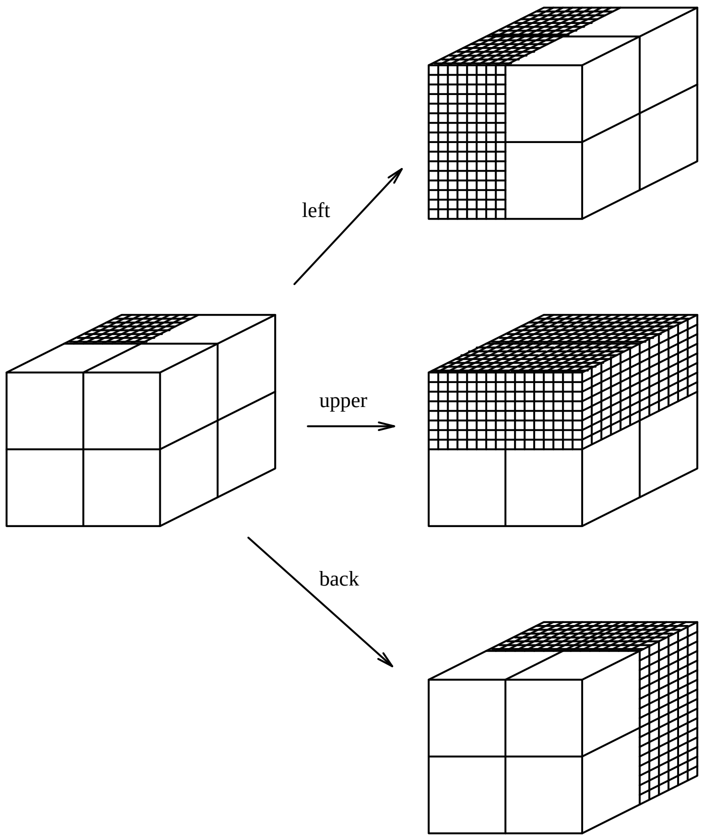

Figure 1: Three independent choices specify 1 box in 8.

stacked up in a cube and you want to find the location of some candy which I put into one of them. You can ask me questions, but I will only answer with "yes" or "no". What is the minimum number of questions that you need to ask? There are poor ways to play this game, and the most obvious way to play it is not the best way. If you point to each box successively and ask "is that it?", you would require an average of 4 questions, since half of the time it would be in the first four boxes and the other half of the time it would be in the second four boxes. (Even if you knew that the candy was in the last box—because the first seven answers were "no"—you would still ask the 8th question when using this method.) In contrast, information theory tells us that you only need $\log_2 8 = 3$ questions:

1. Is it in the left set of four boxes?

2. Is it in the upper set of four boxes?

3. Is it in the back set of four boxes?

Each question divides the set exactly in half, so each yes-no answer gives you 1 bit of information. Notice that the order of the questions doesn't matter and that they could be asked in other ways with the same results. Also, the answer to one question has no effect on the answer to the others since the divisions are at 90° (orthogonal) to each other. We will rely on this *independence* property many times. Every time the number of choices doubles, the number of bits required increases by 1. Thus, if there were 16 boxes, information theory tells us we only need 4 questions, but the "is that it?" approach would use an average of 8.

Our challenge is to understand what the ribosome is doing in terms of the choices made. The problem is that a ribosome doesn't work by cutting the genome in half in a series of steps. Instead, it looks at whole patterns in the RNA by physical touching. Ribosome binding sites are like addresses, so let's label our boxes in binary:

$$000, 001, 010, 011, 100, 101, 110, 111. \tag{1}$$

(See Fig. 2.) If no = 0 and yes = 1, these correspond to the 8 possible answers to the questions listed above. We need three binary digits (*i.e.* bits) to label the 8 boxes. Notice how each question chooses a subset of four of the boxes in the figure. For example, if the first digit is 1, the box is one of the 4 on the left side.

Unlike a human, a ribosome searches by random Brownian motion, and when it comes to the right pattern, it binds and starts translation (Fig. 3). To model this, we have to change the game a little more: I present labeled boxes to you in an arbitrary order and you have to choose the one which has the candy in it. For example, we might agree beforehand that the label 010 means "candy", so you would look for a 010 pattern and when you see this you get to eat the candy inside the box. To make this system work, the proper number of bits has to be written on each box. Clearly 3 bits is the best solution, since 2 bits would be ambiguous and 4 bits would be wasteful.

Now suppose I hid two candies in two different boxes. We can take advantage of the ambiguity when we set the system up. We drop the first question (bit) from each element in list (1) and label the 8 boxes as:

$$00, 01, 10, 11, 00, 01, 10, 11. \tag{2}$$

Candy is in both of the boxes labeled 10. As I randomly present boxes to you, you look for these bits. When you see them, you grab the box and open it for your snack.

You might object at this point, saying that you only need one bit to do the job: simply label the box that has candy 1 and put 0 on each empty box. To address this objection we will make the game even more realistic. Suppose that each box contains a different item,

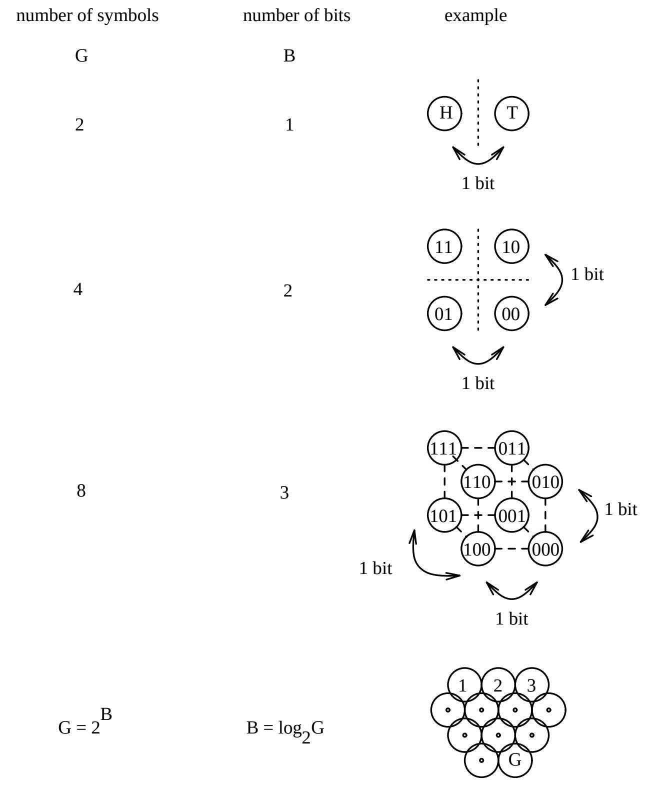

Figure 2: Defining bits as binary choices.
The choice between two equally likely possibilities, such as a coin flip which produces heads (H) or tails (T), is one bit of information. By arranging 4 possibilities into a square, or 8 possibilities into a cube, the independence of binary choices is shown.

---

and we need all of these items at various times. Besides candy, there might be various tools, other kinds of food and so on. Because the boxes have various uses, they need full length labels to be distinguished. This does not mean we have to read the entire label in every case. For example, we could have a code which said that all boxes with food in them begin with 0. In the case of our candies, we can still label each box with 3 bits, but instead of dropping the first bit from the box, we simply ignore it in list (1).

The recognition model that we have made by this step wise process is subtly different from the divide-and-conquer model we began with. The reason we need it is that the ribosome must make choices, but it cannot do so by divide-and-conquer.

Now suppose I have 4.7×10^6 boxes and have put 2600 candies in them. How many bits do you need on each box to identify those which contain candy? This corresponds exactly to the problem the ribosome faces.

For 8 boxes we needed 3 bits. If we doubled the number of boxes to 16, we would need

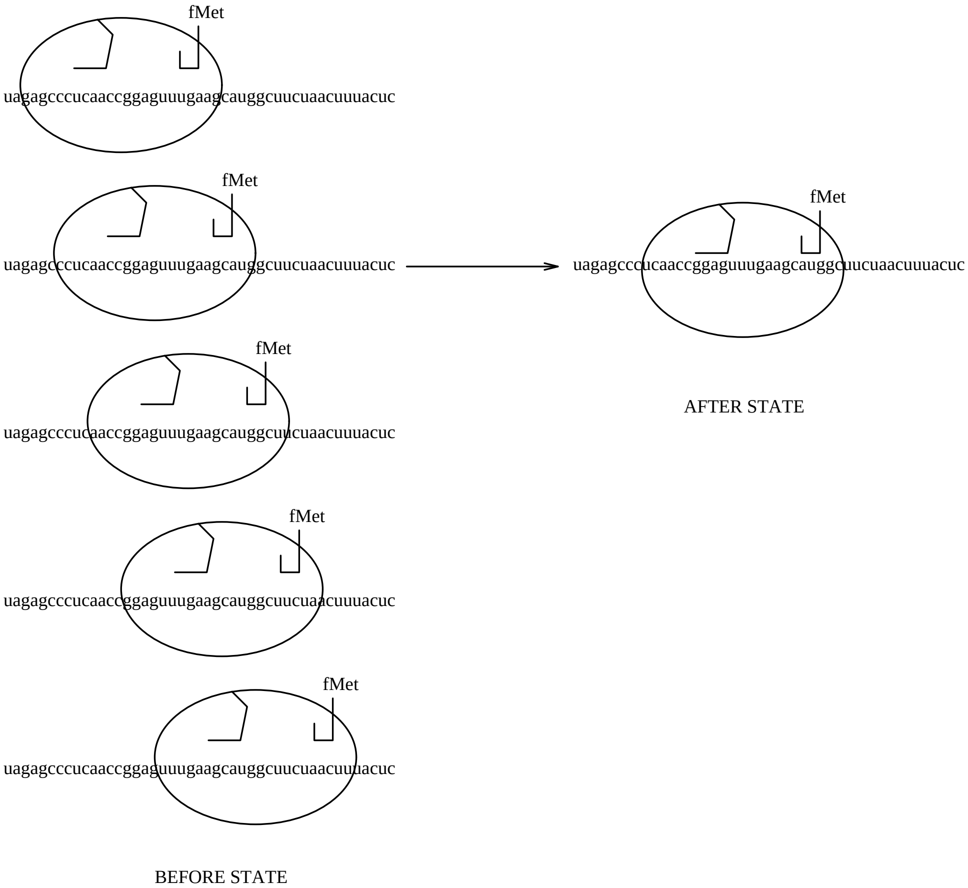

Figure 3: A ribosome chooses one sequence pattern to bind to from many possible patterns. The messenger RNA which the ribosome is searching is shown as a string of a's, c's, g's and u's. This particular sequence is from the coat gene of the MS2 virus (Gold *et al.*, 1981). The ribosome is depicted as an ellipse with two hook-like pieces. The left one represents the $3'$ end of the 16s rRNA, which is the part of the ribosome which recognizes the Shine and Dalgarno sequence (Shine & Dalgarno, 1974), and the right one represents the first transfer RNA to which is attached a formylmethionine (fMet), the first amino acid of the new protein. On the left are shown a few of the states of the ribosome *before* it has found the start site, while on the right is the shown the state *after* the ribosome has found the Shine and Dalgarno ("ggag") and the initiation codon ("aug").

---

one more yes-no question to find a particular box, so a total of 4 bits would be needed. The number of boxes we can choose from, G, grows exponentially as a function of the number of bits, B, according to $G = 2^B$ (Fig. 2). Rearranging this equation to $B = \log_2 G$ we find that the number of bits required grows by the logarithm to the base 2 of the number of boxes.

Now, if 2 of these boxes contained the items you want, you could ignore 1 bit because you wouldn't need to make one decision. If 4 of the boxes had what you want, you could ignore 2 bits; with 8 items you could ignore 3 bits. In this last case, you would get the candy from any box you choose, and you could ignore the entire label. In general, for γ boxes with items in them you can ignore $\log_2 \gamma$ bits. (γ is the Greek letter "gamma".) That is, you only need to look at $\log_2 G - \log_2 \gamma$ bits. We call this number $R_{frequency}$ for two reasons. First, it is directly related to the *rates*, R, of information transmission Shannon was thinking about way back in the 1940's (Shannon, 1948; Shannon & Weaver, 1949): it is the average number of bits needed *per* candy (or binding site) finding operation. Secondly, by rearranging the equation

$$R_{frequency} = \log_2 G - \log_2 \gamma \tag{3}$$

as:

$$R_{frequency} = -\log_2 \frac{\gamma}{G} \tag{4}$$

and we see that γ/G is the *frequency* with which we expect to encounter the items we are looking for during a random search.

For ribosomes, $G = 4.7 \times 10^6$ and $\gamma = 2600$ so the $R_{frequency}$ needed to locate the sites is 10.8 bits per site. Keep this number in mind.

Because we used an exact number of ribosome binding sites (as opposed to saying that some are stronger and others weaker for binding), we have implicitly assumed that these sites are distinct from all other positions on the genome. In Level 1 theory, we will justify this assumption.

Since $R_{frequency}$ depends on the number of sites and the size of the genome, it is fixed by the physiology and genetics of the organism. In contrast, the sequence patterns at the binding sites—corresponding to the labels on our boxes—could have any amount of information independently of $R_{frequency}$. That is, $R_{frequency}$ is a *prediction* about how many bits are needed by the ribosomes to find their sites. Is there enough information at the sites to handle it?

Fortunately for us, many biologists have been busy figuring out the sequences of genes in *E. coli*. In DNA, these consist of long strings of A's, C's, G's and T's, called "bases" or "nucleotides". When a gene is turned on, an RNA copy of these strings is made which differs from DNA in three ways: first, instead of being a double helix, it has only one strand; secondly the sugar in the connecting backbone is ribose instead of deoxyribose and finally "U" is used instead of "T". Ribosomes start translating the RNA at a pattern that has a cluster of G's and A's followed by a gap, and then AUG but sometimes GUG or UUG. In Fig. 4 we show several examples of ribosome binding sites. Notice that at the zero position there are mostly A's but one gene has a G, and that in positions +1 and +2 there are always U's and G's, as we said above. If you look closely you will see the G/A patch in the region around −10.

How much information do we need to describe the patterns here? To say that position +1 always has a U requires telling you 2 bits of information since that is a selection of one

```
    -26  -20    -15   -10   -5    0    +5   +10   +15 +18
     |     |     |     |     |    |    |     |     |   |
 1   AAUUCCAUUCCUAGGAGGUUUGACCUGUGCGAGCUUUUAGUACUC
 2   UAGAGCCCUCAACCGGAGUUUGAAGCAUGGCUUCUAACUUUACUC
 3   UAAGGCAAUGCAAGGUCUCCUAAAAGAUGGAAACCCGAUUCCCUC
 4   CGGCCAUUCAAACAUGAGGAUUACCCAUGUCGAAGACAACAAAGA
 5   GUACUUCACUGAGUAUAAGAGGACAUAUGCCUAAAUUACCGCGUG
 6   CGUUGAAACUUUGGGUCAAUUUGAUCAUGGCAAAAUUAGAGACUG
 7   GUGCUUAGUAACUAAGGAUGAAAUGCAUGUCUAAGACAGCAUCUU
 8   GUUCAAGAUUGCUGGAGGCCUCCACUAUGAAAUCGCGUAGAGGCU
 9   GCUGCUAAAGGUCUAGGAGCUAAAGAAUGGAACAACUCACUAAAA
10   GAAGUUAACACUUUCGGAUAUUUCUGAUGAGUCGAAAAAUUAUCU
```

Figure 4: Ten ribosome binding sites that function in *E. coli.*
The coordinates are given above the sequences. Translation into protein starts at positions 0 through 2 and proceeds to the right. The sequences in this list are from (Gold *et al.*, 1981; Stormo *et al.*, 1982*b*).

---

thing (U) from four things (A, C, G, U). If a position has half A and half G, then that is a selection of 2 from 4, or only 1 bit. In the case of the ribosome we again apply the idea of *before* and *after* states. *Before* binding, the ribosome's "fingers" see 4 possibilities and don't distinguish amongst them. We say that each finger is "uncertain" by $\log_2 4 = 2$ bits. *After* binding, the uncertainty at each finger is lower. If there is 1 base, then the uncertainty is $\log_2 1 = 0$ bits. The decrease in uncertainty is a measure of the sequence conservation or information at the binding site. With 1 base this is $\log_2 4 - \log_2 1 = 2$ bits. If a finger accepts 2 bases *after* then the uncertainty remaining is $\log_2 2 = 1$ bit and the information is $\log_2 4 - \log_2 2 = 1$ bit. When a "finger" accepts all 4 bases, it really doesn't do anything and the information it demands in sequence conservation is $\log_2 4 - \log_2 4 = 0$ bits. When the frequencies are not 0, 50 or 100 percent, a more complex formula is needed to calculate the uncertainty *after* for each finger:

$$H(l) = -\sum_{b \in \{A,C,G,T\}} f(b,l) \log_2 f(b,l) \tag{5}$$

where $f(b,l)$ is the frequency of base b at position l. The derivation and use of the H formula been described in great detail elsewhere (Shannon, 1948; Pierce, 1980; Schneider *et al.*, 1986; Schneider, 1988; Schneider & Stephens, 1990; Stephens & Schneider, 1992; Shaner *et al.*, 1993). Using $H(l)$, we can calculate the information at every position in the site according to:

$$R_{sequence}(l) = 2 - H(l) \qquad \text{(bits per position)}, \tag{6}$$

where we have replaced $\log_2 4$ with the simpler "2". This generates a curve for the information content across the entire binding site. Such curves are difficult to work with because they do not give one a feeling for what the ribosome is looking for. So instead we display those patterns by using a "sequence logo" (Fig. 5), which consists of several stacks of letters. The

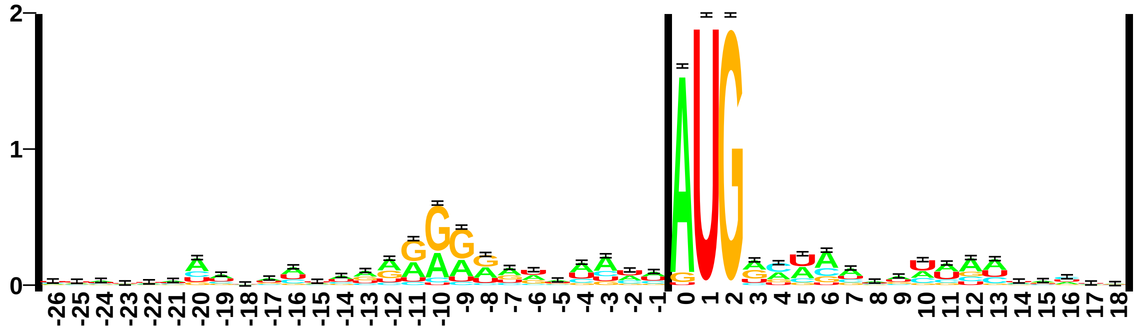

Figure 5: Sequence logo

Sequence logo (Schneider & Stephens, 1990) for 149 *E. coli* ribosome binding sites. The vertical scale is in bits (the black bar is 2 bits high) and the horizontal scale is positions across the sites. Data are from (Schneider *et al.*, 1986). Error bars are for the height of the stack. Coordinates are the same as in Fig. 4. See the text for further details.

---

height of each stack is the information according to equation (6) while within the stack the relative heights of the letters are proportional to their frequencies in the binding site. Notice that the logo instantly shows the features of ribosome binding sites we described above along with many other details. The sequence logo represents the pattern of information, or label, used by the ribosome to indicate "start translation here".

The logarithm is used to calculate information because this gives it a nice property: it is additive if the information sources are *independent*. We can calculate the total amount of pattern in ribosome binding sites simply by summing the information from each position given by equation (6):

$$R_{sequence} = \sum_{l} R_{sequence}(l) \qquad \text{(bits per site).} \tag{7}$$

This is the same as the area under the logo. We get: $R_{sequence} = 11.0 \pm 0.4$ bits per site, which is almost identical to the value of $R_{frequency}$, 10.8, we found earlier![1] *There is just enough pattern at ribosome binding sites* ($R_{sequence}$) *for them to be found in the genetic material of the cell* ($R_{frequency}$). These data imply that there is no excess pattern, and no shortage of pattern. Because $R_{frequency}$ is fixed by physiology and genetics, this result also implies that

[1]The range ± 0.4 was determined by using the Rsim program. See (Stephens & Schneider, 1992).

$R_{sequence}$ must *evolve toward* $R_{frequency}$, a rather subtle result! For further discussion of the evolution of binding sites, see the references (Schneider *et al.*, 1986; Schneider, 1988).

The close proximity of $R_{sequence}$ to $R_{frequency}$ has also been found in other genetic systems (Schneider *et al.*, 1986; Schneider, 1988; Penotti, 1990). There are also apparent exceptions, from which we are learning interesting biology (Schneider *et al.*, 1986; Schneider & Stormo, 1989; Herman & Schneider, 1992; Stephens & Schneider, 1992; Rudd & Schneider, 1992). However, for this paper we want to ask: was it justified to make a model in which ribosome binding sites are completely distinct from everything else? Certainly the $R_{sequence}/R_{frequency}$ data support that model, and it makes good biological sense, but we can go back to Shannon's work to find a theorem which proves it is possible.

3. Level 1 Molecular Machine Theory: Channel Capacity and Precision

When we encounter a noisy phone line, we tend to shout to compensate. This helps us to send more information to the receiver. Likewise, a radio station which has been assigned a restricted range of frequencies by the Federal Communications Commission (in the USA) could use another slice of bandwidth to send more news and entertainment. To describe these relationships, Shannon introduced the concept of channel capacity (C, bits per second). This is a function of bandwidth (W, cycles per second), signal power dissipated at the receiver (P, joules per second) and thermal noise at the receiver (N, joules per second) according to the formula:

$$C = W \log_2 \left(\frac{P}{N} + 1 \right) \qquad \text{(bits per second)} \tag{8}$$

(Shannon, 1948; Shannon & Weaver, 1949). P/N is the so-called signal-to-noise ratio. Shannon proved an amazing theorem about the channel capacity. The first part of the theorem says that if we want to send information at a rate R (bits per second) greater than the channel capacity, ($R > C$) we will fail, and at most the amount C will get through. Noise destroys the rest. (In the worst case, nothing will get through.) The second part of the theorem contains the surprise. If we send at a rate less than or equal to the capacity ($R \leq C$), then we may do so *with as few errors as we may desire*! For example, a communications system to a space probe near Saturn might have only one error in 10^6 bits. According to Shannon's theorem, we could redesign the system to make the error rate one in 10^{12} or even lower. We can't eliminate errors, but we can make them as rare as we want. The way to do this is to encode the signal at the transmitter so that it becomes resistant to thermal noise in the channel. It then must be decoded in the receiver.

Our success so far in applying information theory to biology (*i.e.* Level 0) suggests that the channel capacity results should also apply, but to do this we must translate these ideas into biological terms. We will sketch this translation using the ribosome as an example. Please refer to (Schneider, 1991*a*) for a more rigorous derivation.

Biologists use the analogy of a lock and key to describe the precise interaction of an enzyme and its substrate (Gilbert & Greenberg, 1984). Inside a conventional pin-tumbler lock there is a row of little metal cylinders called "pins" (Macaulay, 1988). In the simplest kind of lock there are two pins, one on top of the other, at each position in the row. The pins can move up and down together in their slots. The combined length of each pair of pins is constant, but the division between them varies up and down. When the proper key is inserted, these divisions are all aligned at the "shear line" and the lower pins can be rotated away from the upper pins. This opens the lock. The wrong key allows one or more pins to prevent the turning. Notice that the pins have to move *independently* for the lock to work properly. Also, if two pins weren't independent then either the lock would never open, or it would be easier to pick it.

Now imagine that the surface of the ribosome has a series of "pins". A representation of the corresponding "key" is shown in Fig. 5. The model we are building says that:

(1) The pins of the ribosome are, to a first approximation, spread linearly along a groove into which the RNA fits.

(2) The pins move independently. (See (Schneider, 1991*a*) and (Stormo *et al.*, 1982*a*) for evidence that this assumption of *independence* is reasonable.)

(3) Each of the three independent directions that a body can move in space corresponds to an independent pin.[2]

(4) Each pin can be modeled by a simple harmonic oscillator.

Thermal vibrations and collisions with neighboring molecules cause the pins to bounce around like weights on springs in the middle of a three-dimensional hailstorm. Most of the time a pin moves slowly, but on occasion it is hit hard by a neighboring molecule and moves quickly until it hits something else or recoils. The total velocity is the sum of many small impacts, so a series of independent velocity measurements would have a bell-shaped Gaussian distribution.

The velocity of an oscillating body in a vacuum follows a simple sine wave, so it has both an amplitude and a phase as independent degrees of freedom. In polar coordinates, the amplitude is represented as the distance from the origin, while the phase is represented by the angle. In polar coordinates the phase and amplitude do not have the same units of measure, but if we convert to rectangular coordinates, we have two numbers which have the same units. (They are actually the two Fourier components of the sine wave.) These two numbers define a point on a plane. If the oscillator is in a thermal bath it is frequently jostled, so a moment later the x and y velocities are slightly different and so the state of the oscillator is represented by a different point. Surprisingly, the distribution of a large collection of such points is circularly symmetric.[3] Three independent descriptive numbers (*i.e.* from $1\frac{1}{2}$ pins) define a

[2] Alternatively, each vibrational mode of a molecule can be thought of as corresponding to a pin. The important point here is that there can be parts of a molecule consisting of groups of atoms which move almost independently from the other groups.

[3] The x and y axis projections of the polar coordinate point are independent and normally distributed: $f(x) = e^{-x^2}$ and $f(y) = e^{-y^2}$. The combined distribution is $f(x, y) = e^{-x^2}e^{-y^2} = e^{-r^2}$, where $x^2 + y^2 = r^2$. This is circularly distributed. See (Schneider, 1991*a*) for further details.

point in a three dimensional "velocity space", and the distribution of many such points is spherical. This distribution is known as the Maxwell gas distribution, since molecules of gas have the same velocity distribution. With four numbers, the distribution becomes a four-dimensional sphere. As we add more pins, we need more numbers to describe their velocities, and the distribution moves into even higher dimensional space. Fortunately the resulting distribution is easy to understand because it is spherical, and its surface becomes more and more distinct as the dimension increases. This is because the total energy in a large molecule fluctuates only a small amount, and the sum of the squares of each velocity component is proportional to the total energy. The square root of that sum is also the distance of each point from the origin in velocity space, so the approximately constant energy means that the points form a thin shell. In really high dimensions it looks like a ping-pong ball in that it has a sharply defined surface. In contrast, each point defines a vector from the center of the space whose direction changes drastically from moment to moment. Every point on this sphere represents an instantaneous configuration of the pin velocities. In the next instant the velocities will be similar, but slightly different so a near-by point will be occupied. As a result, the changing velocities trace a Brownian motion path over the sphere surface, as represented by the squiggling string in Fig. 6. The sphere has a radius proportional to the square root of the temperature, so we can think of it as a "thermal noise sphere".

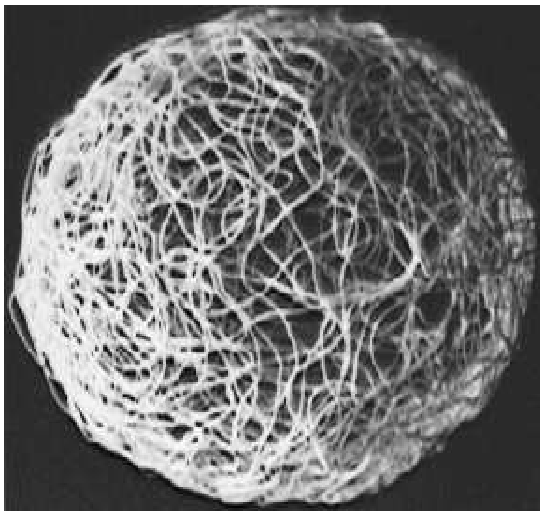

Figure 6: Velocity distribution of a molecular machine.

The surface of a sphere in high dimensional space is modeled by a string squiggling around in 3 dimensional space. Since impacts from neighboring molecules cause the velocities of the pins in a molecular machine to continuously change, the point which they trace on the sphere follows a Brownian motion path. The model was created by dipping string in white glue and layering it onto the surface of a balloon. After the glue dried, the balloon was removed.

The sphere represents the set of all possible velocities of the pins of the ribosome while it is bound to the ribosome binding site waiting for the raw materials with which to begin translation. Therefore it represents the state of the ribosome *after* it has found the sites, and so we call it the *after* sphere. This corresponds to the *after* state shown in Fig. 3.

Recall that one of the ribosome's jobs is to find sites for initiating translation. Before it has done this it is anywhere on the RNA (Fig. 3). (In this model we set aside the problem of finding the RNA.) For the ribosome to stick to its sites, *it must give up some energy to the surrounding medium.* So before being bound to its sites, it must have more energy than after. At the moment the ribosome binds to its sites, the formation of non-covalent bonds converts this potential energy to kinetic energy which is almost immediately dissipated into the surrounding environment, since the ribosome has no way to hold on to it. (As a small molecule surrounded by a huge thermal bath, there is no way for the ribosome to be insulated from the surrounding salt and water medium and the ribosome's kinetic impacts on the surroundings will quickly transfer the excess kinetic energy away.) So before binding, the ribosome has a potential energy with which it *could* attain a higher velocity. Because this energy is released and dissipated during binding, before binding the ribosome is effectively in a larger sphere in the high dimensional velocity space. We call this the *before* sphere. This is a somewhat subtle concept because before binding the ribosome is at the same temperature as the surroundings and after binding it is also at equilibrium with the surroundings. Basically, the *before* sphere reflects the velocities the ribosome *could* obtain from the potential energy it carries before binding. In the same way, we speak of the potential energy of a ball sitting calmly on a table as being equal to the kinetic energy it would have at the moment it hits the floor after having rolled off the edge.

When the machine is in an *after* state, it has energy from the thermal noise, N_y joules. (The subscript y refers to the velocity space, which is called Y-space.) The energy of a mass is proportional to its velocity squared, so $N_y = (r_{after})^2$. (We normalize to remove the proportionality constant.) Turning this around, it means that the radius in velocity space is related to the square root of the energy: $r_{after} = \sqrt{N_y}$. In the *before* state the machine has additional energy P_y joules (which it will dissipate during the machine operation), so the total energy *before* is $P_y + N_y$. That means that the radius of the *before* sphere is $r_{before} = \sqrt{P_y + N_y}$.

Consider a three-dimensional thermal-noise sphere. If we are in the center of the sphere and move north, then there will be two dimensions orthogonal to our direction of motion. Noise can interfere with our motion in both of these directions. In higher dimensions, the geometry of this situation is both weird and interesting; it is shown in Fig. 7. Suppose we have a 100 dimensional space, and we move "north". Then 99% of the interfering noise is orthogonal to our motion. The molecular machine operation can be thought of as the collapse of the *before* sphere with the concomitant movement of the sphere center from point O to B. Since 99% of the noise is at right angles to the line segment between 0 and B, the

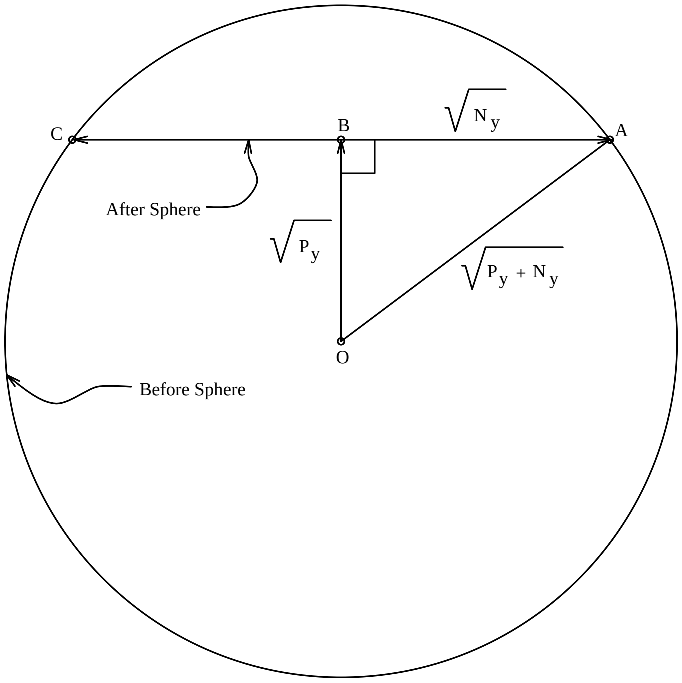

Figure 7: Geometry of high dimensional spheres.

after sphere can be represented by the straight line segment running from C to A. As the dimensionality increases, we may neglect the small amount of noise in the direction O to B. The *after* sphere is still spherical, but must be represented in this diagram by a straight line segment. *Weird!* Now, we know that the *before* sphere has a radius of $\sqrt{P_y + N_y}$ and that the *after* sphere has a radius of $\sqrt{N_y}$ so the Pythagorean theorem applies to show that the distance the sphere center must travel during the collapse is $\sqrt{P_y}$, which corresponds to the energy dissipation P_y. This figure also shows that the *after* sphere always fits inside the *before* sphere for any $P_y > 0$ and $N_y > 0$.

The molecular operation that the ribosome must perform is to select the *after* sphere that represents its binding sites and to avoid all the other *after* spheres that represent different patterns. As we showed above, the *before* sphere encloses all of the *after* spheres (Schneider, 1991*a*). How many choices can the ribosome make for a given energy dissipation? In other words, how many *after* spheres can fit into the *before* sphere? A gumball machine is a good model for the situation (Fig. 8). We can calculate an upper bound for this simply by dividing the volume of the *before* sphere by that of an *after* sphere.

A circle has an area of πr^2, while the volume of a sphere in three dimensions is $\frac{4}{3}\pi r^3$. In a higher dimension, D, the volume is proportional to r^D. Thus the volume of the *before* sphere is $v_{before} \propto r^D_{before} = \sqrt{P_y + N_y}^D$ and that of the after sphere is $v_{after} \propto r^D_{after} = \sqrt{N_y}^D$. The

Figure 8: A gumball machine.
The gumballs represent the possible *after* states of a molecular machine, while the enclosing sphere represents the *before* state. An upper bound on the number of gumballs can be found by dividing the volume of the larger sphere by that of the gumballs. This is the machine capacity.

---

maximum number of small *after* spheres which can fit into the larger *before* is therefore $M = v_{before}/v_{after} = \sqrt{\frac{P_y}{N_y} + 1}^{D}$.

For every pin of the molecule, there are two degrees of freedom of movement (the phase and amplitude projected onto the x and y axes, as mentioned earlier), so the dimensionality $D = 2d_{space}$, where d_{space} is the number of pins. With this we find $M = \left(\frac{P_y}{N_y} + 1\right)^{d_{space}}$.

The base 2 logarithm of the maximum number of choices made is the capacity C_y (in bits), since it represents the largest number of distinct binary decisions the machine could make for the given energy dissipation P_y. To calculate this we take $\log_2 M$, and find the molecular machine capacity:

$$C_y = d_{space} \log_2 \left(\frac{P_y}{N_y} + 1 \right) \qquad \text{(bits per operation).} \tag{9}$$

d_{space} is the number of pins, P_y (joules per operation) is the energy the machine has to give up to go from *before* to *after* and N_y is the thermal noise (in joules) which defines the *after* sphere radius. Notice the resemblance to equation (8). Notice also that the information measures $R_{frequency}$, $R_{sequence}$ and C_y are all different views of the same *before* to *after* state change. Because we use the same sphere geometry that Shannon did (see Appendix 2 of (Schneider, 1991*a*)), we can immediately apply his theorem, which he proved geometrically (Shannon, 1949). When translated into molecular biology and applied to ribosomes, the theorem says:

As long as the ribosome operates at or below its machine capacity, it can be as precise as is necessary for survival.

This theorem shows that it is *possible* for ribosomes to almost completely avoid making the wrong proteins but to still pick out *all* the right ones. To what degree they do this will depend on the evolutionary "pressures" the organism faces. The price that must be paid to obtain this result is that the ribosome must work in a high dimensional space so that the spheres have sharp edges. In other words, the ribosome is forced to have many energetically tiny contacts. This makes ribosome binding sites distinct from other sequences. Since the independent pins of the ribosome appear to be strung out along the RNA, it seems that all those subtle bumps in the sequence logo (Fig. 5) are clues to the code the ribosomes must be using to choose sites.

Notice how messy the ribosome binding site patterns seem to be in Fig. 4 and Fig. 5. This has led people to think that ribosomes could not possibly locate the sites precisely. However, the theory says that the more tiny contacts exist, the more precise the ribosome could become. So the apparent messiness could be, paradoxically, helping the ribosome to be precise!

4. Level 2 Molecular Machine Theory: The New Maxwellian Demons

In 1867, James Clerk Maxwell introduced to the world a little being with a fine touch and a propensity for trouble (Maxwell, 1904). This demon controls a door between two gas vessels, and he uses the door to sort molecules (Fig. 9). When a rapidly moving molecule approachs from the left, he lets it through, but a slow one is not allowed passage. Fast ones from the right don't pass him but slow ones from there do. In this way, fast molecules end up in the right hand container, which becomes hotter, and slow molecules go to the left, which becomes colder. The temperature difference could run a steam engine and supply the world with free energy. The demon is not creating energy from nothing, which would be a First Law violation. Instead, he is violating the Second Law by decreasing the entropy of the system (by doing the separation) without compensatory heat dissipation. Although the demon does nothing more than open and close a door, it seems that one could get a perpetual motion machine that violates the Second Law of Thermodynamics. One of the many forms of this fundamental law (Jaynes, 1988; Atkins, 1984) states that it is not possible to move heat from a body at a lower temperature to one at a higher temperature without performing work. Since it has not been clear what kind of work the demon performs, he has

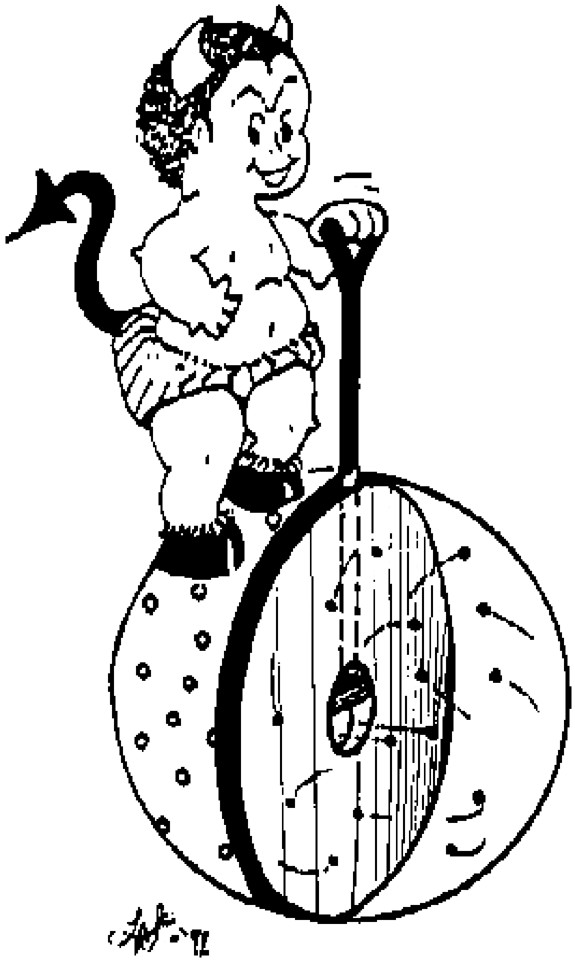

Figure 9: Maxwell's demon.

been causing consternation ever since (Leff & Rex, 1990).

To see the gas molecule, Brillouin supplied the demon with an "electric torch" (Brillouin, 1951). Clearly most of the photons won't bounce off the gas molecules and they would change the energy of the molecules if they did (from the Heisenberg uncertainty principle). Worse, not all of those few that manage to come back to the eye of the demon will be absorbed. For some reason, these and other serious difficulties are usually ignored in the literature, perhaps because the demon is not thought of as being composed of molecules. The Maxwell demon problem is real, but making absurd assumptions only clouds the issues. For example if we assume that there is no thermal noise, as physicists sometimes have, then the machine's capacity goes to infinity according to equation (9). We will try to stick to realistic physics and biology.

What is the demon doing? Let's split the problem into its components. As each molecule approaches he must [1] see it, then he must [2] decide what to do, and finally he must [3] operate the door accordingly. All three of these molecular operations dissipate energy, and they correspond to actions of the molecules of [1] rhodopsin, [2] genetic control elements like the ribosome and *Eco*RI and [3] muscle. These molecular machines all choose substates from several possibilities so they fit the theory described here (Fig. 10).

[1] If it can see, the demon must have rhodopsin (or a rhodopsin-like molecule) because otherwise it is unrealistic. An excited molecule of rhodopsin can flip into a state called bathorhodopsin (Fig. 11). In vertebrate eyes, once a single bathorhodopsin has formed it triggers 400,000 chemical reactions, which then cause a nerve impulse. For this reason, we can see single photons (Lewis & Priore, 1988). All this happens at room temperature, so the molecule operates in the presence of thermal noise. The energy of the photon must be dissipated for the molecule to choose between bathorhodopsin and the original unexcited state, just as a ribosome must dissipate energy to bind to its binding site. Thus rhodopsin is a molecular machine that makes decisions.

[2] The brain of the demon must decide what to do. Ribosomes are an example of genetic control elements which make decisions by choosing their binding sites from among other sequences in the genome.

[3] The same must be true for muscle, in which myosin heads "walk" along actin fibers. Each myosin head must dissipate energy derived from a broken energy molecule—ATP—to choose between its original state and one step forward.

Thus *all* the demon's actions—*seeing, computing* and *movement*—require decisions to be made by molecular machines. The fundamental question now is: does making decisions require using energy?[4] If so, the demon must be given energy to do its job, and the problem is solved.

Surprisingly, the solution comes from the molecular machine capacity formula (9). C_y has units of bits per molecular machine operation. The power P_y has units of joules (dissipated from the machine into the surroundings) per molecular machine operation. Thus we can

[4] At this point, an anonymous reviewer asked why the problem was not phrased in terms of "the resetting of the demon" because this is said to be the important energy dissipating step by people concerned with the limits of computation (Leff & Rex, 1990). To "reset" usually means to set the state of a binary device to 0. That requires choosing one state from two since one could alternatively "reset" to 1. Actually, "resetting" is a particular case of first adding energy to a device and then choosing a sub-state as the energy dissipates into the surroundings. This is what molecular machines do, but the term "reset" generally does not fit well with molecular machines operations. For example, molecular biologists never say that ribosomes "reset" themselves when they bind to their sites. It is more sensible to speak of them as making a choice or decision. Molecular biology forces us to use this more general paradigm.

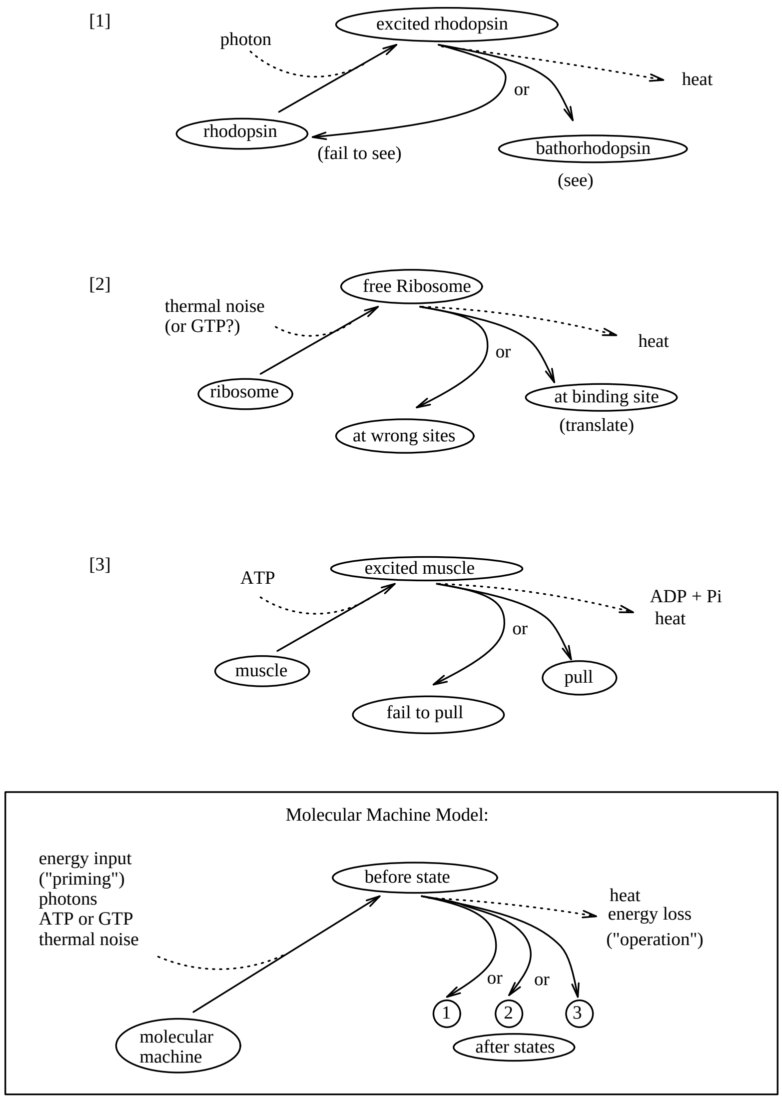

Figure 10: Examples of molecular machines.
All molecular machines are primed by a source of energy from a low energy state into a high *before* state. From the *before* state, the machine may choose one of several alternative *after* states by dissipating the energy into the surrounding medium.

---

define a new variable, the energetic cost:

$$\mathcal{E} \equiv \frac{P_y}{C_y} \tag{10}$$

which has units of joules per bit, just the right ones to answer the question. Now, what

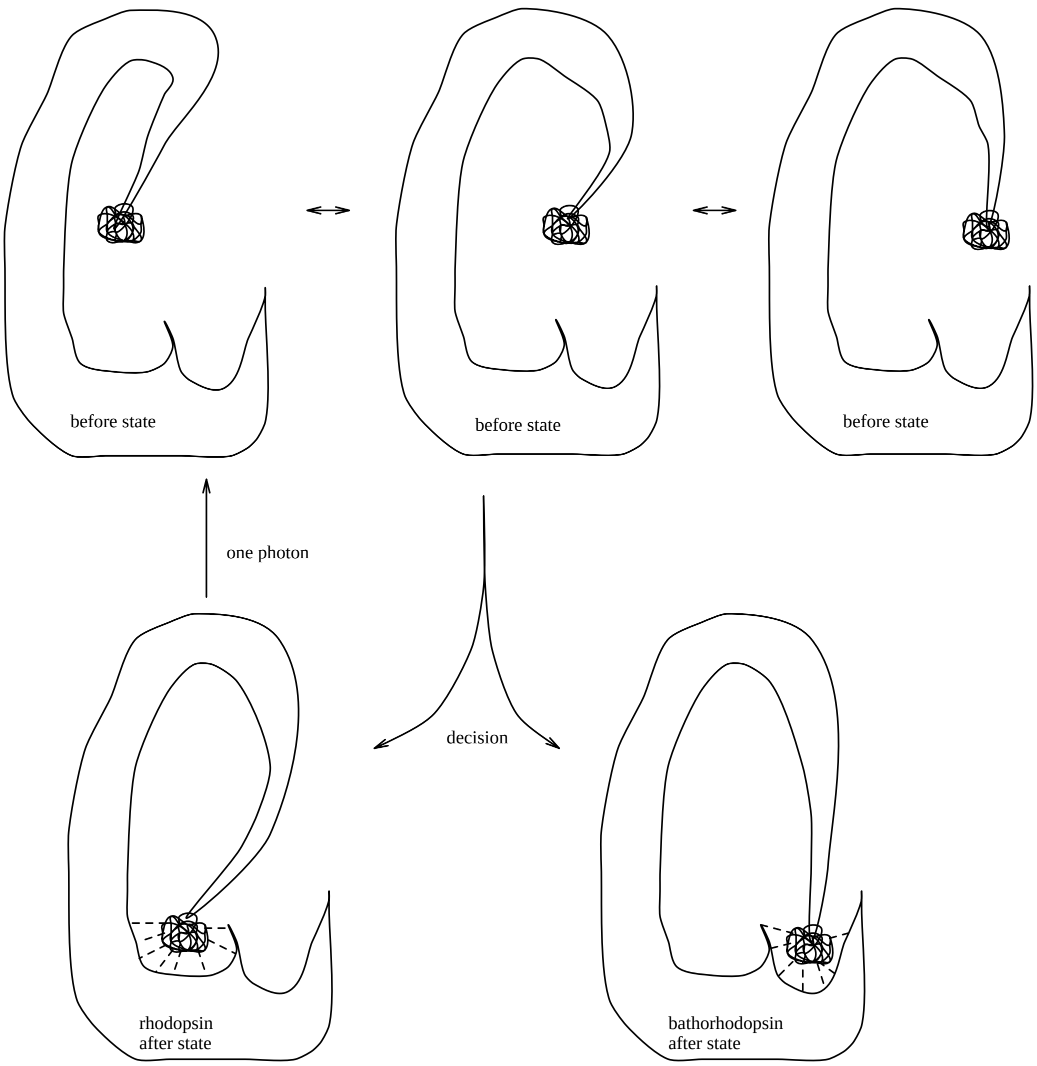

Figure 11: Schematic model of rhodopsin.
A rhodopsin molecule in the dark is portrayed in the lower left corner. The tight ball is bound in a pocket inside the structure of the molecule. Dashed lines indicate weak bonds. When a photon is absorbed by rhodopsin, it passes to an excited state, in which it can move into several configurations (upper row). When the excitation energy is lost, rhodopsin either falls back to its original state (bottom left) or to a bound new state, bathorhodopsin (bottom right). The figure is only intended to show relationships between states according to the molecular machine model. *It does not show the actual shape of rhodopsin.*

---

happens to $\mathcal{E}$ as we reduce the power used by the molecular machine toward zero? Our first thought is that $\mathcal{E}$ also goes to zero, and some people have made this mistake. However, notice that C_y is a function of P_y according to equation (9), and that C_y *also* goes to zero as P_y does. We must take the limit to find the minimum value:

$$\mathcal{E}_{min} \equiv \lim_{P_y \to 0} \mathcal{E} \tag{11}$$

$$= \lim_{P_y \to 0} \frac{P_y}{d_{space} \ln\left(\frac{P_y}{N_y} + 1\right) / \ln(2)} \tag{12}$$

where we have substituted first from equation (10) then from equation (9) and converted to the natural logarithm in preparation for the next step. L'Hôpital's rule (Thomas, 1968) tells us to take the derivative of both top and bottom and try the limit again:

$$\mathcal{E}_{min} = \lim_{P_y \to 0} \frac{\ln(2)\left(\frac{P_y}{N_y} + 1\right) N_y}{d_{space}} = \frac{\ln(2) N_y}{d_{space}} \tag{13}$$

Because we don't know values for d_{space} and N_y, we appear to have reached an impasse, but this is not so. Nyquist showed in 1928 that there is a simple formula for thermal noise (Nyquist, 1928). For every independent degree of freedom in a system, the thermal noise is $\frac{1}{2}k_{\mathrm{B}}T$.

Because the pins in a molecular machine move independently, we can model each of them as a simple harmonic oscillator surrounded by a thermal bath, as we did earlier to construct the thermal noise sphere. Since each oscillating pin has a phase and an amplitude, it has two degrees of freedom. (Note: the phase and amplitude change with time because the oscillator is being bombarded by the surrounding molecules.) Since there are d_{space} independent pins in the machine (by definition), and the number of degrees of freedom is twice this, we can calculate the thermal noise from:

$$N_y = (\tfrac{1}{2}k_{\mathrm{B}}T) \times (2d_{space}). \tag{14}$$

Substituting this into (13) gives

$$\mathcal{E}_{min} = k_{\mathrm{B}}T \ln(2) \quad \text{(joules per bit)}. \tag{15}$$

In electrical engineering power is given in joules per second, so it may approach zero as the energy decreases or as the time taken for dissipation increases. Neither approach alters equation (13), so it is incorrect to think that the limit $\mathcal{E}_{min}$ can be bypassed by slowing the energy dissipation of a device such as a computer. In the case of individual molecules operating at a certain (perhaps irregular) rate per second, the power is given in joules per molecular operation. That is, the energy to be dissipated is the same as the power during one operation and time is irrelevant because we only count complete operations.

Now we start an entirely different approach. To do this it is first important to clarify the relationship between information and entropy. Shannon's information measure is based on the concept of uncertainty (Shannon, 1948). The more possible distinct states that a system has, the higher the uncertainty. Since Shannon required additivity, he found that the uncertainty should be the logarithm of the number of possible states. If the possible states are not equally likely, the formula is:

$$H \equiv -\sum_{i=1}^{\Omega} P_i \log_2 P_i \qquad \text{(bits per microstate)} \tag{16}$$

where P_i is the probability of the i^{th} state, and there are Ω states. We used this formula earlier in equation (5).

If I am sending you a series of characters, you will be uncertain as to what character I will send next *before* you receive each character. *After* receipt, your uncertainty is lower, but not necessarily zero. If there is noise of some kind, it will not affect your uncertainty before receipt, but will affect your uncertainty after. More noise means that less information gets through. Thus the decrease in uncertainty that a communications receiver experiences is the information that the receiver gains. Information is *always* measured as a state function difference. Likewise, the decrease in uncertainty that a molecular machine experiences is the information that the machine gains. That is,

$$R = H_{before} - H_{after} = -\Delta H \qquad \text{(bits per operation)} \tag{17}$$

where R is the information gain. 'R' stands for the 'rate' of information gain, in this case per operation. If there are a given number of operations per second, one can convert to bits per second.

Entropy is another state-function measure of a system. The Boltzmann-Gibbs entropy of a physical system, such as a molecular machine, is

$$S \equiv -k_{\text{B}} \sum_{i=1}^{\Omega} P_i \ln P_i \qquad \left(\frac{\text{joules}}{{}^\circ\text{K} \cdot \text{microstate}}\right) \tag{18}$$

where k_{B} is Boltzmann's constant (1.38×10^{-23} joules / °K) (Waldram, 1985; Weast *et al.*, 1988). Notice the close resemblance to equation (16). The two can be related if we make sure that the probabilities used in the two equations refer to the same states of the molecular machine. By also making sure that we are referring to the same state change:

$$\Delta S = S_{before} - S_{after} \tag{19}$$

any microstates which are not accounted for that do not alter between the two states will cancel. For example, the information needed to choose one thing in two is the same as the information needed to choose two things from four because the extra factor drops out during the subtraction of the logarithms. This solves the problem of relating information theory to thermodynamics, which many writers have complained is not possible. See (Schneider, 1991*b*) for further discussion.

Combining equation (16) with (18) and substituting into (17) and (19) gives:

$$\Delta S = -k_{\rm B} \ln(2) R. \tag{20}$$

This shows that, under the condition that the probabilities refer to the same microscopic states—which is a reasonable assumption in the case of molecular machines—*a decrease in the entropy of the molecular machine corresponds to an increase in the information of that machine.* In other words, *for a molecular machine to make choices, its entropy must decrease.*[5]

Now we are ready to start the second approach to the question of information versus energy use in a molecular machine. It is well known that the Second Law of Thermodynamics can be expressed by this equation:

$$dS \geq \frac{dq}{T} \tag{21}$$

where dS is the entropy increase of a system corresponding to dq, the heat flowing into the system at temperature T (Tolman, 1938; Fermi, 1936; Jaynes, 1988). The system in this case is a molecular machine, and the surrounding is the huge thermal bath of the cell or fluid the machine is in. As we discussed earlier, the molecular machine is completely exposed to the solvent, so its temperature *before* is the same as *after* the operation. Because the temperature is a constant, we can integrate and rearrange:

$$-T\Delta S \leq -q \tag{22}$$

(where $\int dq = q$ by definition since heat is a quantity, not a change). Substituting equation (20) into (22) and rearranging again gives:

$$k_{\rm B} T \ln(2) \leq -q/R \quad \text{(joules per bit)}. \tag{23}$$

If we set $R = 1$, this formula says that for every bit of information gained, the heat dissipated into the surroundings (*i.e. negative* q) must be more than $k_{\rm B}T\ln(2)$. This is a new way to think about the Second Law! (See (Jaynes, 1988) for a clear exposition on the amazing variety of forms of this law.)[6]

[5] If $H_{after} = 0$, equation (17) collapses. Unfortunately, this has fooled many authors into thinking that equation (16) gives the information. Because of equation (18), they then think that entropy is proportional to information. Thus a measure of "disorder" becomes associated with "information" and the confusion is complete. To avoid this widespread error, always measure information as a decrease in uncertainty.

[6] Landauer's statement that "throwing away information requires dissipation" (Landauer, 1991) contradicts this form of the Second Law. Suppose we have a binary device which can be either in a 0 or a 1 state. Let us define "erase" and "reset" to mean to set the device to 0, starting from either the 0 or 1 state. Since

Let us define the energetic cost as the heat given up for the information gained:

$$\mathcal{E} \equiv -q/R \tag{24}$$

so that equation (23) becomes:

$$\mathcal{E} \geq k_{\mathrm{B}}T\ln(2) \quad \text{(joules per bit).} \tag{25}$$

The smallest possible value of $\mathcal{E}$ can be called $\mathcal{E}_{min}$ so we now have this form for the Second Law of Thermodynamics:

$$\mathcal{E}_{min} = k_{\mathrm{B}}T\ln(2) \quad \text{(joules per bit).} \tag{26}$$

Surprise! This is the same formula we derived from the channel capacity, equation (15)! Since we took a limit to obtain this, the channel capacity is a more general law than the Second Law under isothermal conditions.[7]

Because the capacity formula gives a precise bound according to Shannon's theorem, we can see that $\mathcal{E}_{min}$ also defines a *precise* limitation on the capabilities of a molecular

the device is stable in its initial state, whether 0 or 1, we must add some energy to it to alter that state. This is called "priming" the machine (Schneider, 1991*a*). From the excited state, the device is guided by pre-set inputs to fall into the 0 state, and hence it must dissipate energy to the surroundings. For example, to change the state of a coin requires that we add some energy to rotate it. Then the coin can only come to rest again by giving up its rotational energy. (Do not be fooled into thinking that a coin in the 0 state need not go through this process because "you" can see it is in that state and save yourself the trouble. Remember: you are not involved in the actual operation of molecular machines, and even if you were, your seeing, thinking and acting require energy dissipation! Specifically, to know that the state is 0 requires a reading operation.) Thus this second step of erasure looks exactly the same as selecting a sub-state, which is equivalent to an information gain of the device, by the precise definition we use (Schneider *et al.*, 1986; Schneider, 1991*b*). This is not really so odd as it may seem, since we must do exactly the same thing to set the device to 1. "Erasing" requires the same process as filling a memory with whatever information patterns we choose. It is easy to demonstrate this—simply switch the labels on each flip-flop so that they all register zero. Thus "erasing" consists of two steps: priming and setting. *Information in a memory is lost at the priming step, and not at the energy dissipation step.* This is a direct restatement of the Second Law because energy must be *absorbed* by the memory to destroy the information there. The energy is then dissipated to set the device and gain different information. Landauer's confusion came about because he lumped the priming and the setting operations into a single step, and because he did not take care to rigorously define what he means by "information". We can avoid the confusion by always describing these steps explicitly, and by defining "information" as a difference in the uncertainty state function (Schneider, 1991*b*). Similar arguments apply even when the two states are at different energy levels, as in a transistor (Mead & Conway, 1980).

[7] The fundamental importance of equation (26) is not widely recognized. Authors often loosely use the phrase "$k_{\mathrm{B}}T\ln(2)$" or even "approximately $k_{\mathrm{B}}T$". This leads to two difficulties. First, using disconnected mathematical phrases has led people to overlook the idea that they are part of an important equation. Further, *this equation has units*: joules per bit. If one chooses base 10 for measuring choices, one obtains $\mathcal{E} \geq k_{\mathrm{B}}T\ln(10)$ joules per *digit*. For making calculations, it is necessary to keep the units in mind.

machine. Equations (15) and (26) say that for every bit of decision Maxwell's demon makes, it must dissipate at least (1 bit) $\times\, \mathcal{E}_{min} = k_{\mathrm{B}}T\ln(2)$ joules of energy into the surrounding environment. It can dissipate more, but it cannot dissipate less, since to do so would violate both Shannon's channel capacity theorem and the Second Law. To preserve the First Law, the energy used to prime the machine into its *before* state must come from somewhere else, so the demon must use at least $\mathcal{E}_{min}$ joules to see the gas molecule, another $\mathcal{E}_{min}$ joules to decide which kind of molecule it is, and *another* $\mathcal{E}_{min}$ joules to choose the proper door configuration.[8] We might imagine then that if the demon only passes molecules with energy more than $3\mathcal{E}_{min}$ joules then there would be a net gain in energy. But a molecule with $3\mathcal{E}_{min}$ joules of energy above background will be more rare than one with less energy. In a thermal bath at equilibrium (which is the situation the Demon starts with, and is a better condition than the situation later on after he has been working for awhile), the molecules follow a Boltzmann distribution in which the probability p of a molecule with energy ΔE above background is

$$p = e^{-\Delta E/k_{\mathrm{B}}T}. \tag{27}$$

Letting $\Delta E = 3\mathcal{E}_{min} = 3k_{\mathrm{B}}T\ln(2)$, we find that $p = 1/8$. That is, the demon must reject 7 molecules for every one it allows to pass. But to do this it must make $log_2(8) = 3$ bits of choice and so by the form of the Second Law given here, he must *spend* at least $3\mathcal{E}_{min}$ joules of energy. It's a no-win situation!

Another possibility is a "pressure" demon who captures molecules from one side of a membrane and puts them on the other. Such pumps are found on the surface of all cells, and they use ATP to do their work. Once the molecule is captured, this demon must decide which side to put it on. This 1 bit decision costs at least $k_{\mathrm{B}}T\ln(2)$ joules of energy. The result is the evacuation of one chamber and the filling of the other, so the pressure difference could run an engine. Classical thermodynamics tells us that if we allow the molecules to spread out again by expanding the volume of a piston, the most work we could do with them is $k_{\mathrm{B}}T$ multiplied by the logarithm of the ratio of the volumes before and after. For doubling the volume (a 1 bit loss in the positional uncertainty of the gas molecules), the formula is $k_{\mathrm{B}}T\ln(2)$, which is, of course, $\mathcal{E}_{min}$, so this kind of demon can't win either.

The problem with Maxwell's demon has been that he is imaginary, so it is easy to forget an important influence on his performance, such as thermal noise. By making our model of the demon match molecular machines found in cells, we can avoid this difficulty. To do this, we have split the operations of the demon into three simpler steps well known in biology: seeing, thinking and moving. For clarity, we further divided each of these steps into priming and choosing (Fig. 10). Having dissected the little bugger, let's see if he's still alive.

The Second Law was formulated because nobody could find a way around its various constraints on heat and work. No known violations exist, even at the molecular level (McClare, 1971). Some people think that molecules can somehow bypass the Second Law because

[8]These steps could be linked to save energy.

they are so small. If thermodynamics and statistical mechanics don't apply to molecular machines, then a single molecule could violate the Second Law of Thermodynamics. If one machine can violate it, then so could a large number of them acting in parallel. It is clear that this is not happening since living things depend heavily on the operations of molecular machines, yet they all need to eat energy molecules or photons.

The main question arising from Maxwell's demon is whether the Second Law of thermodynamics can be violated by some kind of subtle sorting operation. The new form of the Second Law given by equation (26) shows that for every bit of sorting that the demon performs, he must pay at least $\mathcal{E}_{min}$ joules of energy dissipation away from himself. He cannot win because that separation costs at least as much as it would be worth, as shown above. Thus the demon cannot violate the Second Law by doing sorting operations. But sorting is the only trick that the demon has available to him! *The demon violates the Second Law by decreasing the entropy of a system without compensatory dissipation.* Maxwell's demon is dead.

The elimination of the demon from our discussion of these issues will not be a tragic loss. The demon has been like the proverbial angels-on-a-pinhead in that no two people agree on how to define the issues involved. This has led to a huge and confused literature (Leff & Rex, 1990). We can clarify the problem of molecular sorting by replacing the imaginary demon with examples of real molecular machines. These biologically active molecules have the wonderful advantage that we can play with them in the laboratory to see if our models are reasonable (Schneider & Stormo, 1989).

5. Molecular Computers

The most important requirement for building a computer is a way to perform Boolean logic, since this allows one to construct logic circuits. Many molecules perform the AND function. For example, the *Eco*RI enzyme effectively asks whether

(base #1 is G) AND (base #2 is A) AND (base #3 is A) AND
(base #4 is T) AND (base #5 is T) AND (base #6 is C)

Genetic control systems often work by one molecule binding to a spot to prevent another molecule from binding there. This is a NOT operation. AND followed by NOT is a NAND, and *all* logical circuits can be constructed from NAND (Wait, 1967; Gersting, 1986; Schilling *et al.*, 1989). The theory of molecular machines shows that molecular machines can act as precisely as we choose, so nearly ideal Boolean logic could exist and complete molecular circuits should be possible to build. Thus we already know enough about molecular biology to know that molecular computers are possible.

These machines would have to dissipate at least the quantity of energy required by the Second Law of Thermodynamics, as given by equation (15). Fortunately this is only a tiny quantity compared to today's computers (Keyes, 1988), so molecular computers should be compact and energy efficient.

One of the confusing issues here is that the dissipation of a computer need not be correlated to the number of Boolean operations we write down. (See (Leff & Rex, 1990) for a review of "non-dissipative" computing, and (Gonick, 1991*a*) for an introduction to circuit analysis.) Following the convention that $\cdot$ means AND, and that $+$ means OR, we can create a circuit with 4 inputs and 1 output from the equation

$$X = (A \cdot C + A \cdot D) + (B \cdot C + B \cdot D) \tag{28}$$

as shown in Fig. 12a. Each of the 7 gates makes a choice between 0 and 1, so the total

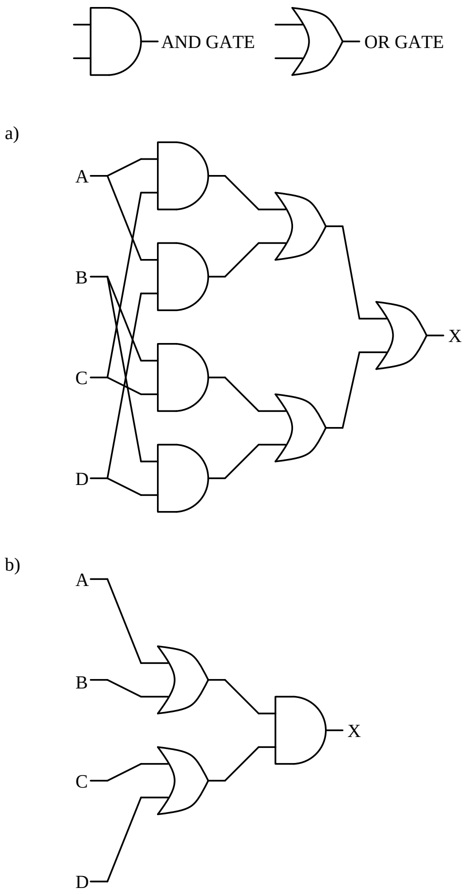

Figure 12: Equivalent Boolean circuits.

energy dissipation of the circuit *as drawn* would be at least $7\mathcal{E}_{min}$ joules. However, we can simplify equation (28) to:

$$X = (A + B) \cdot (C + D) \tag{29}$$

The equivalent circuit uses only 3 gates as shown in Fig. 12b, so this circuit would dissipate only $3\mathcal{E}_{min}$ joules.

We can do *even better* than this. What really counts is the number of choices the circuit makes (Feynman, 1987), its "output". Since Boolean variable X has only two possibilities, setting it must dissipate at least $1\mathcal{E}_{min}$ joules. As Feynman pointed out, we can't avoid this dissipation.

But there is an additional cost which Feynman did not identify. Setting the input states A, B, C and D takes another $4\mathcal{E}_{min}$ joules no matter what the circuit is. So the total minimum dissipation cost is $5\mathcal{E}_{min}$ joules because we have to count both the total number of inputs *and* outputs to find the cost of the operation.[9] This is the lower limit for this circuit according to the Second Law. Such a circuit could operate like a key in a lock, with all computations being done in one single concerted step.

The discussions in the literature point out that we can perform arbitrary amounts of "computation" for little energy dissipation. There are two ways to see that this is both true and not necessarily as great as it sounds. First, ribosomes must "compute" the fact that they are not at a binding site, and since they don't stick in this case, they don't dissipate energy. Thus if we put a ribosome with some RNA that does not have a ribosome binding site, it would "compute" forever, with no dissipation at all. Of course it would never get to translate, so this situation is not profitable for a cell. The second way to see that huge amounts of computation are not necessarily so good is to notice that this seems to imply that it would be better to build the more complex circuit shown in Fig. 12a, which computes 7 boolean operations, rather than the equivalent simple one shown in Fig. 12b, which computes 3 boolean operations. That would, of course, waste building materials. Even larger amounts of computation could be had for even larger and more absurd circuits. The idea that infinite amounts of computation can be performed without any dissipation costs is therefore correct, but performing such extra computations (while keeping the same input and output) is just wheel spinning. The fundamentally required energetic cost is at the input and output of a computation. Memory must also choose states, so it too entails this cost.

Can we perform dissipationless computation? Surprisingly, this is done all the time in molecular biology labs around the world. We call them "Southerns", "Northerns", "Westerns" and even "Far Easterns".[10] To perform a Southern, a molecular biologist will purify some DNA (from, for example, a person) then cut the DNA with a restriction enzyme such as *Eco*RI, which cuts between the G and A of the DNA sequence GAATTC. In the test tube there are many copies of the genetic material, and the enzyme chops up each piece at exactly the same places. This mixture is then put onto one end of gel, which is like a big slab of jello between two sheets of glass. A voltage applied to the gel forces the DNA to run between the sheets. Little pieces of DNA zip through fastest, while bigger pieces get tangled

[9] Feynman only counted output gates.

[10] The originator of the method was E. M. Southern (Southern, 1975); the later names are whimsical.

up in the gel and take longer. As a result, the *Eco*RI fragments separate by size inside the gel. Before the pieces run off the end, the gel is then removed and placed on top of some filter paper. The filter paper is put on top of some dry paper towels. Wet paper towels are put on top of the gel. Water runs through the gel and is absorbed in the dry towels. This transfers the DNA from the gel to the filter paper. The filter is removed and then baked to fix the DNA onto it. Next, a single piece of DNA called a "probe" is made radioactive. This is put in a plastic bag with the filter and some buffer. The probe is everywhere in the bag, but it will stick only to the DNA molecules on the filter to which it is complementary. (A pairs to T and C pairs to G.) The filter is washed and placed against film. Wherever the probe can pair to the filter-bound DNA, a black band appears on the film. In this way one can, for example, pick out single genes from the entire genetic material.

Inside the bag, each probe DNA tumbles in solution "computing" whether or not to bind to the genetic DNA. Only when it binds does a dissipation occur. Thus fixing the output of the "Southern" computer required dissipation. When the probe DNA was made, its sequence (choice of the order of the letters A, C, G, T) was determined. Making this choice required an energy dissipation. That is, setting the input to the "Southern" computer also requires energy dissipation. "Westerns" use the same idea, but the radioactive probe is an antibody, and the material spread out by size is a set of proteins. "Northerns" use RNA spread instead of DNA, and "Far Easterns" use a dried gel instead of a transfer to filter paper. Only the input and output of these methods dissipates energy, the "computation" itself doesn't.

Imagine a futuristic technology in which we had a molecule with a series of switches on its surface. We set the state of the molecule, dissipating energy to do so. We then release it on one end of a reaction chamber. The molecule then diffuses randomly by Brownian motion to the other side of the chamber where there is a bank of many kinds of receptors, each one representing a possible answer to the computation we are trying to perform. If the switch states of the released molecule match the surface of receptor, then the molecule binds and triggers an "answer". The entire computation is performed in a single concerted step. Input and output require energy dissipation, but the computation in between is "dissipationless". Such an advanced molecular computer is actually not so far fetched! We have just described a process similar to the hormonal communications in our bodies.

But how can *we* build molecular computers? Molecular biology allows us to construct any protein we can write down on paper. If we could build and connect together NAND gates, we could construct entire molecular computers. The reason we don't have them yet is that we don't know how to design proteins to fold properly, to bind to specific surfaces, or to use energy.

Fortunately, in the past few years a path to the construction of molecular computers has opened:

1. *Stolen Parts*. Modern molecular biology allows us to take pieces from nature and to hack together logic circuits like those in genetic control systems. This can be done

now!

2. *Evolved Parts.* Powerful evolutionary techniques such as SELEX (Tuerk & Gold, 1990) and the creation of viruses which present variants of binding proteins on their surfaces (Smith, 1985; Scott & Smith, 1990; Delvin *et al.*, 1990) should allow us to select the interacting surfaces we need to put molecular parts together or to construct logic circuits. But because these evolutionary methods work by guesses, and optimal solutions may be rare, this may be an arduous task.

3. *Designed Parts.* Because different codes correspond to packing the spheres together in different ways, the theory of molecular machines shows that the way to make molecules select their substates correctly is to code their structure properly. A long term challenge is to understand the coding used by molecular machines deeply enough that we can design them ourselves. The so-called "genetic code" is only one of many, since every molecular machine operation must involve a coded interaction, and there are many kinds of molecular machines in the world. Using the tools of molecular biology, we may choose to study any of these. Fortunately we already have 40 years of experience with coding theory (Sloane, 1984). Once we have broken the codes of natural molecular machines, we will have opened the door to a fully engineered nanotechnology.

Acknowledgments I thank John Spouge, Mark Shaner, Mort Schultz, Peter Lemkin, Stanley Brown, Denise Rubens, Jake Maizel and Paul Hengen for useful conversations and for comments on the manuscript; April Cruz for the method of creating Fig. 6; Janice Currens, Jon Summers and Annie Chaltain for photography; and Linda Engle for her drawing of Maxwell's Demon.

References

Atkins, P. W. (1984). *The Second Law.* W. H. Freeman and Co., N. Y.

Berg, O. G. & von Hippel, P. H. (1987). Selection of DNA binding sites by regulatory proteins, statistical-mechanical theory and application to operators and promoters. *J. Mol. Biol.* **193**, 723–750.

Berg, O. G. & von Hippel, P. H. (1988). Selection of DNA binding sites by regulatory proteins. II. The binding specificity of cyclic AMP receptor protein to recognition sites. *J. Mol. Biol.* **200**, 709–723.

Bradley, D. (1993). Will future computers be all wet? *Science,* **259**, 890–893.

Brillouin, L. (1951). Maxwell's daemon cannot operate: information and entropy. I. *J. of Applied Physics,* **22**, 334–337.

Day, I. N. M. (1992). Analysis of the 5′-AAUAAA motif and its flanking sequence in human RNA: relevance to cDNA library sorting. *Gene,* **110**, 245–249.

Delvin, J. J., Panganiban, L. C. & Delvin, P. E. (1990). Random peptide libraries: a source of specific protein binding molecules. *Science,* **249**, 404–406.

Drexler, K. E. (1981). Molecular engineering: an approach to the development of general capabilities for molecular manipulation. *Proc. Natl Acad. Sci. USA,* **78**, 5275–5278.

Drexler, K. E. (1986). *Engines of Creation.* Anchor Press, Garden City, New York.

Eiglmeier, K., Honoré, N., Iuchi, S., Lin, E. C. C. & Cole, S. T. (1989). Molecular genetic analysis of FNR-dependent promoters. *Mol. Microb.* **3**, 869–878.

Fermi, E. (1936). *Thermodynamics.* Dover Publications, Inc., New York.

Feynman, R. P. (1961). There's plenty of room at the bottom. In *Miniaturization*, (Gilbert, H. D., ed.), pp. 282–296, Reinhold Publishing Corporation, New York.

Feynman, R. P. (1987). Tiny computers obeying quantum mechanical laws. In *New Directions in Physics: The Los Alamos 40th Anniversary Volume*, (Metropolis, N., Kerr, D. M. & Rota, G., eds), pp. 7–25, Academic Press, Inc., Boston.

Fields, C. (1990). Information content of *Caenorhabditis elegans* splice site sequences varies with intron length. *Nucleic Acids Res.* **18**, 1509–1512.

Gersting, J. L. (1986). *Mathematical structures for computer science.* second edition, W. H. Freeman and Co., New York.

Gilbert, S. F. & Greenberg, J. P. (1984). Intellectual traditions in the life sciences. II. Stereocomplementarity. *Perspect. Biol. Med.* **28**, 18–34.

Gold, L., Pribnow, D., Schneider, T., Shinedling, S., Singer, B. S. & Stormo, G. (1981). Translational initiation in prokaryotes. *Annu. Rev. Microbiol.* **35**, 365–403.

Gonick, L. (1991*a*). *The Cartoon Guide to Computers.* second edition, HarperCollins, New York, NY.

Gonick, L. (1991*b*). *The Cartoon Guide to Genetics.* updated edition, Barnes & Nobel, New York, NY.

Herman, N. D. & Schneider, T. D. (1992). High information conservation implies that at least three proteins bind independently to F plasmid *incD* repeats. *J. Bacteriol.* **174**, 3558–3560.

Jaynes, E. T. (1988). The evolution of Carnot's principle. In *Maximum-Entropy and Bayesian Methods in Science and Engineering*, (Erickson, G. J. & Smith, C. R., eds), vol. 1, pp. 267–281, Kluwer Academic Publishers, Dordrecht, The Netherlands. http://bayes.wustl.edu/etj/articles/ccarnot.ps.gz http://bayes.wustl.edu/etj/articles/ccarnot.pdf.

Keyes, R. W. (1988). Miniaturization of electronics and its limits. *IBM J. Res. Dev.* **32** (1), 24–28.

Kohara, Y., Akiyama, K. & Isono, K. (1987). The physical map of the whole E. coli chromosome: application of a new strategy for rapid analysis and sorting of a large genomic library. *Cell,* **50**, 495–508.

Landauer, R. (1991). Information is physical. *Physics Today,* **44** (5), 23–29.

Leff, H. S. & Rex, A. F. (1990). *Maxwell's Demon: Entropy, Information, Computing.* Princeton University Press, Princeton, N. J.

Lewis, A. & Priore, L. V. D. (1988). The biophysics of visual photoreception. *Physics Today,* **41**, 38–46.

Macaulay, D. (1988). *The Way Things Work.* Houghton Mifflin Company, Boston.

Mars, M. & Beaud, G. (1987). Characterization of vaccinia virus early promoters and evaluation of their informational content. *J. Mol. Biol.* **198**, 619–631.

Maxwell, J. C. (1904). *Theory of Heat.* Longmans, Green and Co., London.

McClare, C. W. F. (1971). Chemical machines, Maxwell's demon and living organisms. *J. Theor. Biol.* **30**, 1–34.

Mead, C. & Conway, L. (1980). *Introduction to VLSI Systems.* Addison-Wesley Publishing Company, Reading, Mass.

Nyquist, H. (1928). Thermal agitation of electric charge in conductors. *Physical Review,* **32**, 110–113.

Papp, P. P., Chattoraj, D. K. & Schneider, T. D. (1993). Information analysis of sequences that bind the replication initiator RepA. *J. Mol. Biol.* **233**, 219–230.

Penotti, F. E. (1990). Human DNA TATA boxes and transcription initiation sites: a statistical study. *J. Mol. Biol.* **213**, 37–52.

Penotti, F. E. (1991). Human pre-mRNA splicing signals. *J. Theor. Biol.* **150**, 385–420.

Pierce, J. R. (1980). *An Introduction to Information Theory: Symbols, Signals and Noise.* second edition, Dover Publications, Inc., New York.

Pingoud, A. (1985). Spermidine increases the accuracy of type II restriction endonucleases. Suppression of cleavage at degenerate, non-symmetrical sites. *Eur. J. Biochem,* **147**, 105–109.

Polisky, B., Greene, P., Garfin, D. E., McCarthy, B. J., Goodman, H. M. & Boyer, H. W. (1975). Specificity of substrate recognition by the *Eco*RI restriction endonuclease. *Proc. Natl Acad. Sci. USA,* **72**, 3310–3314.

Rosenberg, J. M., McClarin, J. A., Frederick, C. A., Wang, B. C., Boyer, H. W., Grable, J. & Greene, P. (1987). The structure and function of *Eco*RI endonuclease. In *Biological Organization: Macromolecular Interactions at High Resolution*, (Burnett, R. M. & Vogel, H. J., eds), pp. 11–43, Academic Press, Inc., Orlando.

Rudd, K. E. & Schneider, T. D. (1992). Compilation of *E. coli* ribosome binding sites. In *A Short Course in Bacterial Genetics: A Laboratory Manual and Handbook for* Escherichia coli *and Related Bacteria*, (Miller, J. H., ed.), pp. 17.19–17.45, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, New York.

Schilling, D. L., Belove, C., Apelewicz, T. & Saccardi, R. J. (1989). *Electronic circuits, discrete and integrated.* third edition, McGraw-Hill, New York.

Schneider, T. D. (1988). Information and entropy of patterns in genetic switches. In *Maximum-Entropy and Bayesian Methods in Science and Engineering*, (Erickson, G. J. & Smith, C. R., eds), vol. 2, pp. 147–154, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Schneider, T. D. (1991*a*). Theory of molecular machines. I. Channel capacity of molecular machines. *J. Theor. Biol.* **148**, 83–123. http://www.lecb.ncifcrf.gov/˜toms/paper/ccmm/.

Schneider, T. D. (1991*b*). Theory of molecular machines. II. Energy dissipation from molecular machines. *J. Theor. Biol.* **148**, 125–137. http://www.lecb.ncifcrf.gov/˜toms/paper/edmm/.

Schneider, T. D. & Stephens, R. M. (1990). Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res.* **18**, 6097–6100. http://www.lecb.ncifcrf.gov/˜toms/paper/logopaper/.

Schneider, T. D. & Stormo, G. D. (1989). Excess information at bacteriophage T7 genomic promoters detected by a random cloning technique. *Nucleic Acids Res.* **17**, 659–674.

Schneider, T. D., Stormo, G. D., Gold, L. & Ehrenfeucht, A. (1986). Information content of binding sites on nucleotide sequences. *J. Mol. Biol.* **188**, 415–431.

Scott, K. K. & Smith, G. P. (1990). Searching for peptide ligands with an epitope library. *Science,* **249**, 386–390.

Shaner, M. C., Blair, I. M. & Schneider, T. D. (1993). Sequence logos: a powerful, yet simple, tool. In *Proceedings of the Twenty-Sixth Annual Hawaii International Conference on System Sciences, Volume 1: Architecture and Biotechnology Computing*, (Mudge, T. N., Milutinovic, V. & Hunter, L., eds), pp. 813–821, IEEE Computer Society Press, Los Alamitos, CA. http://www.lecb.ncifcrf.gov/˜toms/paper/hawaii/.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Tech. J.* **27**, 379–423, 623–656. http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html.

Shannon, C. E. (1949). Communication in the presence of noise. *Proc. IRE,* **37**, 10–21.

Shannon, C. E. & Weaver, W. (1949). *The Mathematical Theory of Communication.* University of Illinois Press, Urbana.

Shenkin, P. S., Erman, B. & Mastrandrea, L. D. (1991). Information-theoretical entropy as a measure of sequence variability. *Proteins: Structure, Function, and Genetics,* **11**, 297–313.

Shine, J. & Dalgarno, L. (1974). The 3′-terminal sequence of *Escherichia coli* 16S ribosomal RNA: complementarity to nonsense triplets and ribosome binding sites. *Proc. Natl Acad. Sci. USA,* **71**, 1342–1346.

Sloane, N. J. A. (1984). The packing of spheres. *Sci. Am.* **250** (1), 116–125.

Sloane, N. J. A. & Wyner, A. D. (1993). *Claude Elwood Shannon: Collected Papers.* IEEE Press, Piscataway, NJ.

Smith, G. P. (1985). Filamentous fusion phage: novel expression vectors that display cloned antigens on the viron surface. *Science,* **228**, 1315–1317.

Southern, E. M. (1975). Detection of specific sequences among DNA fragments separated by gel electrophoresis. *J. Mol. Biol.* **98**, 503–517.

Stephens, R. M. & Schneider, T. D. (1992). Features of spliceosome evolution and function inferred from an analysis of the information at human splice sites. *J. Mol. Biol.* **228**, 1124–1136. http://www.lecb.ncifcrf.gov/˜toms/paper/splice/.

Stormo, G. D. (1990). Consensus patterns in DNA. *Meth. Enzym.* **183**, 211–221.

Stormo, G. D., Schneider, T. D., Gold, L. & Ehrenfeucht, A. (1982*a*). Use of the 'Perceptron' algorithm to distinguish translational initiation sites in *E. coli*. *Nucleic Acids Res.* **10**, 2997–3011.

Stormo, G. D., Schneider, T. D. & Gold, L. M. (1982*b*). Characterization of translational initiation sites in *E. coli*. *Nucleic Acids Res.* **10**, 2971–2996.

Thomas, G. B. (1968). *Calculus and Analytic Geometry.* fourth edition, Addison-Wesley, Reading, Mass.

Tolman, R. C. (1938). *The Principles of Statistical Mechanics.* Dover Publications, Inc., New York.

Tuerk, C. & Gold, L. (1990). Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. *Science,* **249**, 505–510.

Wait, J. V. (1967). Symbolic logic and practical applications. In *Digital Computer User's Handbook*, (Klerer, M. & Korn, G. A., eds), pp. 4–3 to 4–28 show that NAND is sufficient, McGraw-Hill Book Company, Inc., N. Y.

Waldram, J. R. (1985). *The Theory of Thermodynamics.* Cambridge University Press, Cambridge.

Watson, J. D., Hopkins, N. H., Roberts, J. W., Steitz, J. A. & Weiner, A. M. (1987). *Molecular Biology of the Gene.* fourth edition, The Benjamin/Cummings Publishing Co., Inc., Menlo Park, California.

Weast, R. C., Astle, M. J. & Beyer, W. H. (1988). *CRC Handbook of Chemistry and Physics.* CRC Press, Inc., Boca Raton, Florida.

Woodhead, J. L., Bhave, N. & Malcolm, A. D. B. (1981). Cation dependence of restriction endonuclease *Eco*RI activity. *Eur. J. Biochem,* **115**, 293–296.