First version: *February 12, 2006* Last update: *April 25, 2012*

A gentle introduction to…

INFORMATION CONTENT IN TRANSCRIPTION FACTOR BINDING SITES

Summary

The application of Information Theory to the analysis of genomic sequences and, specifically, transcription factor binding sites, is one of the most successful attempts at modeling the behavior and evolution of these genomic elements. As such, information theory models lie at the core of many transcription factor binding site search and motif discovery tools, such as the widely used Virtual Footprint and MEME. Nonetheless, the use of information theory methods has some caveats that are not usually taken into account explicitly. Many of these caveats are usually ineffectual, but they be a significant in certain cases, in the same way as a default minimum word-size in BLAST can have unanticipated consequences.

With BLAST, the main problem faced by uninitiated users is that they need to understand the particulars of the algorithm in order to understand what the effects of parameter changes, like minimum word-size, might be. The algorithmic basis of BLAST may not be very difficult to understand, but the algorithm has a relatively large number of fine-tuning tweaks that can easily befuddle any user. In the case of Information Theory-based methods, algorithms are kept relatively simple, yet the assumptions on which they rely and the whole concept of Information Theory can be difficult to grasp or just plain counterintuitive to uninitiated users. This manuscript has been written to provide a paced introduction to Information Theory as applied to transcription factor binding sites. It does not seek to be a complete reference to the field of information theory or to its application in biology. On the contrary, readers are encouraged to explore the cited references and other freely available introductions to the subject.

Please send feedback and comments to: erill@umbc.edu

INDEX

Why information?

Life is, in essence, an information process. Even though nobody adheres completely to a single definition of life, the concept of autopoiesis (Varela, Maturana et al. 1974) (a self-contained set of processes devoted to sustain themselves) is a common tenet of most life definitions. Autopoiesis and the related concept of homeostasis (the art of maintaining a system in a stable condition despite changes in the environment) are obvious examples of information processing: the system must respond to external stimuli in order to adapt; often by changing internal states. Moreover, autopoiesis implies that the bounded system must continually exploit external resources to self-sustain itself; otherwise, the second law of thermodynamics would prevail and eventually disrupt the system. Therefore, an autopoietic system must be able to self-sustain by allowing a net flow of information and energy through it. This is again clearly reminiscent of life as we know it. Animals, for instance, eat *information* (highly ordered organic compounds) and excrete *randomness* (much less ordered organic compounds)[1].

The other sense in which life is an information process is the unifying concept of evolution by natural selection, which encompasses even some structures that defy the autopoietic interpretation of life (e.g. viruses). Life as we know it evolves by natural selection. This is a hugely massive information process: mutations arise by chance and are either removed (typically) or preserved (much less frequently) by natural selection (some are also fixated by genetic drift, but this is a limitation, rather than a fundamental property, of the underlying information process). Evolution by natural selection is an extremely wasteful method: millions of individuals must die due to purifying selection (they are removed from the population because they carry harmful mutations) if selection is to operate properly and make a thorough exploration of the genetic landscape. Thus, it is only ironic, yet proof of its power, that such an inefficient method leads to extremely efficient autopoietic systems, such as the human body.

Why regulation?

Just as any other information processing system, such as a computer, living beings operate on a program. In this case, the program must ensure their perpetuation through life (autopoiesis) and their eventual reproduction against an ever changing environment consisting of natural phenomena and

[1] This is, at best, a rough analogy, since information (as considered here) and physical entropy are only tangentially related. It is true, however, that the maintenance and processing of stable information within a living being requires an internal low-entropy state to be kept moderately constant, and that this in turn requires an energy expenditure or the acquisition of organic matter in low entropy states. For more about this, see Tillman, F. and B. Roswell Russell (1961). "Information and entropy." Synthese **13**(3): 233-241.

other living beings. Even though life has some unique properties that computers have yet to reach, such as encoding hardware and software simultaneously on a single information-carrying molecule, the basic mechanisms of information processing remain the same as in computers: life needs a set of instructions/operations and extensive programming (see (Robbins 1992) for a more detailed and instructive comparison between living beings and computers). Roughly, and at the risk of stretching the analogy to far, we can equate life's instructions with genes (which encode proteins or RNA capable of performing some biochemical function). Regulation, on the other hand, equates with computer programming.

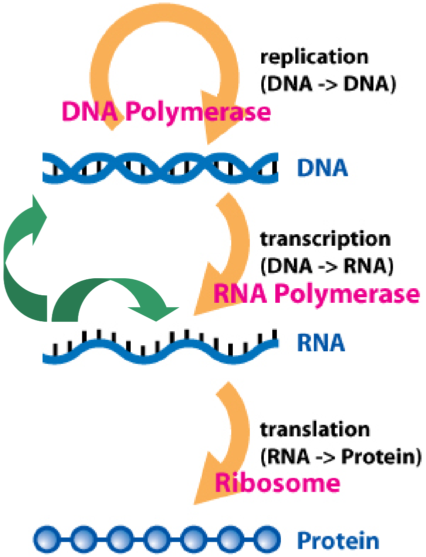

Figure 1 – Information flow following the Central Dogma of Molecular Biology. Original image by Daniel Horspool. Notice that sometimes information does not flow into protein; ribosomal RNA (rRNA), a structural component of ribosomes, is the end product of the information flow for the genes encoding ribosomal RNA. Notice also (in green) that in viruses, RNA can replicate itself using RNA-dependent RNA polymerases. Likewise, retroviruses like HIV can use reverse transcriptase to transfer information from RNA to DNA.

It is becoming more and more widely accepted that differences in regulation, as opposed to differences in the number or type of genes (i.e. in the instruction set), are the main contributors to the phenotypic variation we see around us. Our protein coding gene sequences (i.e. instructions) may be 99% identical to those of chimps, but we surely do not feel 99% identical to chimps, nor 35% identical to daffodils (most of the times, that is) (Marks 2002). So, where are the differences? It turns out that many large phenotypic differences can be explained by small changes in regulation (i.e. programming), especially in organisms that follow developmental *programs*. We can draw here again on a relatively hazy parallel with computer science, where programming can be seen as low (machine) level (e.g. assembler), very high level (Java, Python) or in between (C, C++). As it turns out, regulation also takes place at all levels of the information processing that occurs at the molecular level (Table 1). This information process is summarized by the Central Dogma of Molecular Biology: DNA is **transcribed** into RNA (typically called messenger RNA, mRNA), which is then (often after some processing) **translated** into a chain of amino acids, which eventually fold into a functional protein (Figure 1).

Stage	Mechanism	Mechanism
Pre-transcription	Gene copy number	Gene location
Transcription	Transcriptional activation	Transcriptional repression
Transcription	Polymerase subunits	Polarity
Post-transcription	micro RNA (miRNA)	RNA interference (RNAi)
Post-transcription	mRNA lifetime	Leader peptides
Post-transcription	Alternative splicing	mRNA capping
Translation	Number/position of RBS	Codon usage
Translation	Frame-shifting	Stop-codon read-through
Post-translation	Proteolysis	Phosphorylation
Post-translation	Chaperones	Multi-protein assemblies

Table 1 – Non-inclusive list of regulatory mechanisms at different stages of the molecular information process. Information theory has been typically applied to transcriptional regulation, but can also be applied to other regulatory mechanisms.

Transcriptional regulation is the most evident, best understood and probably the most common mechanism of regulation. It involves the activation or repression of gene transcription (DNA→mRNA) and it is usually accomplished by binding of proteins to the DNA sequence, in a region upstream of the regulated gene called the *promoter* region. Regulating things at the transcriptional level makes the most intuitive sense. Transcription is the first flexible real-time process of gene expression. Therefore, regulation before transcription cannot be very flexible and regulation in later stages will always be more wasteful (because transcripts, at the least, will have been produced no matter what).

There is no clear definition for what a gene or a promoter is (e.g. some people might exclude the RBS from the promoter while others would not)[2]. Nonetheless, the basic mechanism of operation for transcriptional regulation involves binding of a number of proteins (called transcription factors) to DNA in the "promoter" region. In order to *transcribe* a gene, the RNA-polymerase enzyme must first bind to the promoter region (either directly or indirectly, through co-factors). By binding to DNA in the promoter region, transcription factors may either activate *transcription* (assisting RNA-polymerase) or repress it (most often by literally blocking the binding site of RNA-polymerase or some of its co-factors).

[2] This is a rather convoluted issue. I tend to include the ribosome binding site (RBS) in a protein coding gene as part of the gene (and of the promoter), since it is a fundamental unit required for the proper translation, albeit not for the transcription, of the gene product.

In fact, some transcription factors are able to repress some genes and activate others (Collado-Vides, Magasanik et al. 1991).

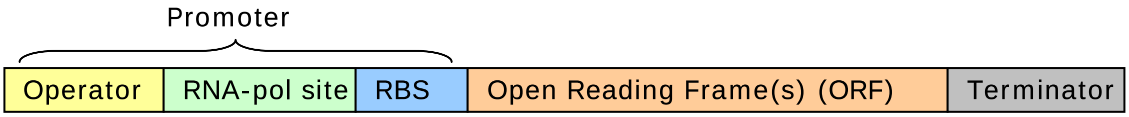

Figure 2 – Basic structure of a "gene" in molecular biology. The "gene" is composed of an Open Reading Frame (ORF) that will be *translated* into protein (not in the case of an RNA gene). A Ribosome Binding Site (RBS) allows ribosomes to attach to the *transcribed* mRNA molecule and start *translation*. A terminator is responsible for disengaging RNA-polymerase (the enzyme responsible for *transcribing* DNA into mRNA) from the template DNA. A binding site for RNA-polymerase allows this enzyme to attach to DNA and start *transcription*. A protein, called a transcription factor, binds the operator sequence and regulates the rate of *transcription* by assisting/hindering the activity of the RNA-polymerase enzyme.

Transcription factor binding sites and motifs

Even though some transcription factors bind to DNA in a very unspecific way, or seem to recognize tridimensional properties of the DNA molecule (e.g. curvature) rather than specific sequences, most transcription factors bind to DNA by recognizing specific sequence elements, called transcription factor (TF) binding sites. Typically, a transcription factor recognizes not just one particular sequence (e.g. ATGGACCAT), but a number of similar sequences. This collection of slightly different sequences, and its diverse set of representations, is collectively known as the **binding motif** (or binding pattern). An example of sequences recognized by a transcription factor would be the following list of 25 binding sites for the hypothetical protein BUH.

ATGACATCAT	ATTCGCTAAT	ATTGCGAGAT	GTGTGATCAT	ATGTTGCCAG
ATGCGACAAT	GCTAGCTCAG	ATGCTGATAT	GTACTGACAT	ATGAGATTAT
ATGCTGCCAA	TAGCTAGCAT	TTGTGATGAT	ATGCATTCAG	ATCAGACCAT
ATGCGATAGG	ATCGCGCCAT	TTAGCATGCC	ATGAATACTT	ATGACAGCAT
ATCGACGTAC	ATCGCTACAT	ATTGCATCAG	ATGGACCCCT	ATGATGACTT

Table 2 – List (or collection) of binding sites for the hypothetical protein BUH.

A classical representation of the above binding motif is the consensus sequence, in this case the sequence ATGCGATCAT. The consensus sequence is derived as the sequence containing the most frequent base at each position of the binding motif. The consensus sequence might be a useful (and compact) representation of the collection of binding sites bound by a given protein, but it is not a very robust representation (Schneider 2002). For instance, given only the ATGCGATCAT consensus you would be *equally* likely to assume that ATG**AC**ATCAT, ATG**TA**ATCAT and **GG**GCGATCAT are members of the BUH collection of binding sites, since they all match the consensus sequence at 8 positions. However, if you take a quick look at the BUH collection, you will probably realize that only the first site

(ATG**AC**ATCAT) is in fact a member, while the second (ATG**TA**ATCAT) could easily be. However, it will soon become rather obvious that the third site (**GG**GCGATCAT) is not likely to be a BUH binding site.

Representation of binding motifs

The reasoning behind your intuitive ranking of these sites lies in probability theory and in the fact that AT in the first two positions is very frequent (and GG very infrequent), making the **GG**GCGATCAT motif highly unlikely, no matter how well it matches the rest of the consensus. This leads us to a frequentialist representation of the binding motif, in what is typically known as a Position Specific Frequency Matrix (PSFM).

	1	2	3	4	5	6	7	8	9	10
A	0.76	0.04	0.08	0.28	0.12	0.44	0.24	0.12	0.80	0.04
C	0.00	0.04	0.12	0.32	0.28	0.12	0.28	0.68	0.08	0.04
T	0.12	0.92	0.16	0.16	0.28	0.12	0.40	0.08	0.08	0.68
G	0.12	0.00	0.64	0.24	0.32	0.32	0.08	0.12	0.04	0.24

Table 3 – Position Specific Frequency Matrix for transcription factor BUH.

Each cell of the PSFM is the observed frequency p(S_l) of base *S* (*S*∈[A,T,G,C]) at position *l* in the binding motif. The PSFM is a more explicit representation of the binding motif than the actual list of sites, even though a bit more bulky than a consensus sequence. If we examine it under the light of the PSFM, it is quite easy to tell why **GG**GCGATCAT is a bad choice for a BUH binding site: only 12% of the sites contain a G in the first position, and *none* contain a G in the second position[3], so the odds of **GG**GCGATCAT being a BUH site are really low given the information at hand.

In fact, the odds of **GG**GCGATCAT being a site can be computed simply by transforming the PSFM into a Position Specific Weight (or Scoring) Matrix (PSWM). If we assume a background frequency *f(S)* of 0.25 for all four DNA bases, then we can compute the log likelihood ratio as:

$$\log_2\left(\frac{p(S_l)}{f(S)}\right) \qquad (1)$$

for each cell of the PSFM, leading to the PSWM:

[3] A 10^{-100} term (or pseudocount) has been artificially added to all frequencies in order to avoid log(0) problems in further computations. This leads to the -330.19 value in Table 4, instead of -∞.

	1	2	3	4	5	6	7	8	9	10
A	1.60	-2.64	-1.64	0.16	-1.06	0.82	-0.06	-1.06	1.68	-2.64
C	-330.19	-2.64	-1.06	0.36	0.16	-1.06	0.16	1.44	-1.64	-2.64
T	-1.06	1.88	-0.64	-0.64	0.16	-1.06	0.68	-1.64	-1.64	1.44
G	-1.06	-330.19	1.36	-0.06	0.36	0.36	-1.64	-1.06	-2.64	-0.06

Table 4 – Position Specific Weight Matrix for transcription factor BUH. Yellow cells indicate the path followed to obtain the score of the putative **GG**GCGATCAT site.

Plotting (yellow) the path of a sequence like **GG**GCGATCAT along the matrix and assuming positional independence (we just add the scores at each position), we obtain a global score of -323.11. In contrast, ATG**TA**ATCAT yields a score of 9.2 and the sequence ATG**AC**ATCAT, a real member of the collection, gives 11.22.

Both PSFM and PSWM are numeric representations of a binding motif. When the number of positions is large, or when the same principle is applied to protein sequences (with 20 (amino acids), instead of 4 (bases), rows in the matrix), both PSFM and PSWM can become cumbersome. Currently, the preferred way of reporting binding motifs is through Sequence Logos (Schneider and Stephens 1990), which you can generate on the fly using several available web servers [1,2,3,4,5] (Crooks, Hon et al. 2004).

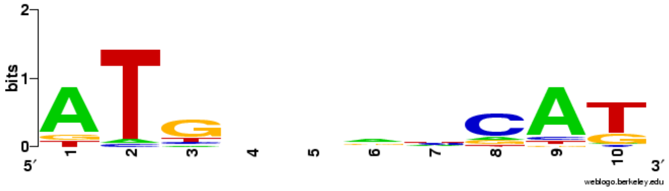

Figure 3 – Sequence logo for the binding motif BUH, as generated by weblogo.berkeley.edu, using the collection in **Table 2** as input. Notice that the Y-axis units are bits.

The sequence logo is a great invention, as it graphically depicts the frequency information available in the PSFM. We can easily see that anything starting with ATG and ending with CAT is likely to be a binding site for BUH and that, for this very reason, **GG**GCGATCAT is really a bad BUH binding site. But the sequence logo is not a direct translation of the PSFM. A graphical representation of the PSFM would be the following frequency plot:

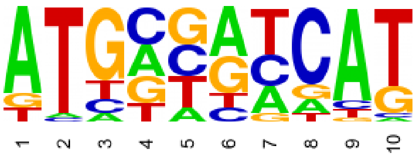

Figure 4 – Frequency plot for the BUH binding motif (generated through http://genome.tugraz.at/Logo/) using the collection in **Table 2** as input.

Clearly, the sequence logo is not only giving us information on the frequency of each base at each position. It is actually giving us also information on the degree of *conservation* of each motif position. Crucially, we can notice that the Y-axis units are **bits**. To many, the word *bit* rings a bell, although for the wrong reason here. *Bit* is an acronym for *BInary digiT* used extensively in computer sciences. In computer jargon, 8 bits classically constitute a *byte*, which is the basic unit we use when measuring the size of our new hard disk (with a hefty Greek modifier as prefix; these days the fashionable prefix being Tera [for as large as to be *monstrous* – from Greek teras/terata: monster]). So a *bit*, in this sense, is a discrete storage unit with two possible values (0/1).

Bits, however, are also a unit of the Mathematical Theory of Information (or Information Theory for short), so the question that begs an answer here is: what has, if anything, Information Theory (IT) to do with proteins and binding motifs? Why are we using bits in our sequence logo of BUH?

On information and information theory

Information theory might be called, more accurately, the mathematical theory of the *measurement* of information, since it does not deal with a wide semantic definition of information, but rather with an abridged, more precise definition that allows us to *quantify* information and, most often, *measure* its *transmission*. Thus, even though information can be conceived of without an observer (much as the noise of a tree falling in an uninhabited forest), in practice information is always coupled with a *sender* (or *source*), a *receiver* and some means of communication: a *channel*.

Information, surprise and uncertainty

The concept of information is inherently linked with uncertainty and surprise. If we attend a bioinformatics symposium and the speaker tells us that he has a PhD in Biology, we would tend to say that he is not giving us a lot of information. This is because we already suspect that he might be a biologist, so his message has not greatly increased our information about him. Or, restating the same, his message has not *surprised* us, because we were expecting something like it. Or to restate it even more: our uncertainty regarding his professional training was quite low, so his message does not decrease our uncertainty much. If, on the contrary, the speaker confesses that he is a proficient player of the caisa drum, he is giving us a lot of information; we clearly could not deduce that much from his being a speaker in a bioinformatics symposium! In other words, our uncertainty about what the speaker does in his leisure time was maximal. Therefore, his confession about playing the caisa takes us completely by surprise and gives us a lot of information (or to rephrase it more correctly, it decreases significantly our uncertainty about what he does in his leisure time).

The bioinformatics speaker example also illustrates another crucial aspect of information theory: its outright detachment from semantics. When we talk about the *amount* of information we receive from the speaker, it does not matter whether we are interested in his leisure time or not. Even if we could not care less about the speaker's leisure time and we were very interested in what his training background was, he *would still be* giving us more information when he confesses to playing the caisa than when he tells us he has a PhD in Biology. We are surprised by the former, but not so much by the latter, because we were much more uncertain about the former than about the latter. And this is true even if we don't care about the former and we do care about the latter.

Information and additivity

Given the relationship exposed above between information, surprise and uncertainty, we could decide to define information as a degree of *decrease in uncertainty*. For instance, before we roll a true die, our initial uncertainty about the result is 6, since there are 6 possible outcomes and all are equally likely. Once we roll the die and observe the result, our uncertainty is zero. Therefore, we can say that in the process of rolling the die and observing the result we have gained an amount of information equal to 6. This simplistic approach has some pitfalls. Suppose now that we roll a die and we toss a coin. We have 6 outcomes for the die, 2 for the coin, so 12 possible paired results. Likewise for two dice: 6 outcomes each, 36 possibilities. So, following the above approach, we could say that we have gained 36 units of information after rolling two dice. This, however, does not ring true to many information processes we observe in the world. Intuitively, the information you gain (or the decrease in uncertainty you experience) after rolling two dice and observing the result is twice the information you get from rolling one, not six times the original information. In other words, information *feels* additive. You would say you have learned approximately twice as many things from two different books (without content overlap), not a figure proportional to the concepts in one book *times* the concepts in the other.

Measuring information

A very suitable function for measuring the additivity of uncertainty in regard to information is the logarithm, since $\log_a(X\cdot Y) = \log_a(X) + \log_a(Y)$[4]. This is precisely, the definition of information introduced by Hartley in 1928 (Hartley 1928):

$$I(N) = \log(N) \qquad (2)$$

where *N* is the number of possible results of a certain experiment.

Implicit within Hartley's formulation is the idea that *I(N)* is the decrease in uncertainty we will experience upon observing the result of the experiment, when our uncertainty will become effectively zero. That is, *I(N)* is measuring our *a priori* uncertainty about a particular result of the experiment, which is the maximum amount of information we can exact from observing that same result.

Hartley's information measure is useful because it relates to our intuitive grasp of information in that: (1) information increases with the number of possible results *N* of an experiment (e.g. rolling a die) and (2) information is additive regarding the combination or repetition of experiments [$\log(n\cdot m) = \log(n)+\log(m)$; $\log(n^m) = m\cdot\log(n)$].[5]

Hartley's definition of information is useful for as long as we consider equiprobable results in our experiments, but it would obviously not work for biased dice. If, for instance, I roll a dice biased so that 6 turns up 50% of the time, it seems evident that the information a 6 provides us with is less than that provided by a 3 (as 3 is a more unexpected result than 6). This leads us to incorporate the probability of each possible outcome in our definition of information. And, again, this is precisely what Claude Shannon did in his seminal paper "A Mathematical Theory of Communication" (Shannon 1948), when he created and formalized the field of Information Theory as we know it today.

$$I(a_i) = \log\left(\frac{1}{P(a_i)}\right) = \log(1) - \log\big(P(a_i)\big) = -\log\big(P(a_i)\big), \quad P(a_i) \in [0,1] \qquad (3)$$

[4] See Schneider's Information Theory Primer (Schneider, T. D. (1995). Information Theory Primer.) for a neat introduction to logarithms and their rules in the Primer's appendix.

[5] Using the logarithm as the function for information/uncertainty does not only make intuitive sense. As seen in the "questions game" below, the logarithm *$\log_x(N)$* corresponds to the number of questions that we need to ask to identify a specific outcome out of *N* possible events with *x* equiprobable outcomes. This is beautifully illustrated in layman terms, along with many other concepts relating to entropy, by Ben-Naim (Ben-Naim, A. (2007). Entropy demystified : the second law reduced to plain common sense. Hackensack, N.J., World Scientific.; see also http://www.ariehbennaim.com/simulations/simulation6.htm for an online simulation of the guessing game)

Shannon’s definition of information *I(a_i)*, following Hartley’s, makes a lot of sense. When the probability of an outcome (a_i) is high, ($P \rightarrow 1$) the information we stand to get from observing that outcome is low ($-\log(\sim 1) \rightarrow 0$). In contrast, when the probability of an outcome is low ($P \rightarrow 0$), the information we stand to obtain increases drastically ($-\log(\sim 0) \rightarrow +\infty$). For the equiprobable case, Shannon’s definition becomes Hartley’s:

$$I(a_i) = -\log\left(\frac{1}{n}\right) = -\left[\log(1) - \log(n)\right] = \log(n), \quad P(a_i) = \frac{1}{n} \tag{4}$$

As with Hartley, it must be stressed that here Shannon is computing information as the implicit difference between *a priori* uncertainty about a particular result of the experiment and complete certainty after observing that exact result.

Information units

We got into this discussion of information because we saw that sequence logos had bits as units in their Y-axes. We said then that bits were no mere BInary digiTs, but units of Information Theory. So, what are bits exactly in the information theory sense?

If we want to measure information, it makes sense to use a unit to measure it with, and logic dictates that this unit should be the simplest possible. If we think of the simplest experiment possible, one with a single, unique outcome, we find that it cannot convey information: we always know the outcome beforehand. If we have two possible results, however, and if we assume that they are equally probable, we obtain:

$$I(a_1) = I(a_2) = -\log\left(\frac{1}{2}\right) = \log(2) \tag{5}$$

If we use base 2 for the logarithm, we obtain $\log_2(2)=1$, and we say that this amount of information is one **bit**. In other words: *a* bit *is the maximum amount of information conveyed by an experiment with two equally probable outcomes*. Or, in other words, *a* bit *is the decrease in uncertainty we experience when observing the result of an experiment with two equally probable outcomes*.

Since the unit of information stems from the base applied to the logarithm, one can define other units based on other logarithm bases. For instance, a *nat* is the information conveyed by an experiment with *e* (~2.71828) equiprobable outcomes, while a *dit* is the information conveyed by an experiment with 10 equiprobable outcomes. Moving from one unit to the other is just a matter of changing the base of the logarithm:

$$\log_a(x) = \frac{\log_b(x)}{\log_b(a)} = \frac{1}{\log_b(a)} \cdot \log_b(x) \qquad (6)$$

$$I(a_i) = -\log_{10}\left(\frac{1}{10}\right) = \log_{10}(10) = 1\ dit$$

$$I(a_i) = \frac{1}{\log_{10}(2)} \cdot \log_{10}(10) = \frac{1}{0.301} \cdot 1 = \log_2(10) = 3.32\ bits$$

so that one *dit* corresponds to 3.32 *bits*. Which makes sense, since we need a little bit more than 3 binary digits to encode ten numbers (3 binary digits lead only to 8 numbers [`000, 001, 010, 011, 100, 101, 110, 111`] since 2^3=8).

On entropy

Up to now we have concerned ourselves solely with the information conveyed by a particular outcome of an experiment. Working in the field of communications (at Bell Labs), Shannon was not thinking about experiments like rolling dice and their repetition or combination with other experiments like tossing coins. Instead, for Shannon the outcomes of experiments were the possible *symbols* emitted by a *source*. He was interested in analyzing the *average information* emitted by a source.

More formally, given a memory-less[6] source *X* that emits messages using a given alphabet $S=[a_1,\ldots,a_n]$ with emission probabilities (for each symbol a_i of *S*) $P=[p_1,\ldots,p_n]$, Shannon defined the entropy of the source *X* as:

$$H(X) = -\sum_{i=1}^{N}\left(p_i \cdot \log(p_i)\right) = \sum_{i=1}^{N}\left(p_i \cdot I(a_i)\right), \quad N = number\ of\ source\ symbols \qquad (7)$$

Therefore, *H(X)* is the weighted mean, or mathematical expectation, of $-\log(p_i)=I(a_i)$. In other words, *H(X)* is the mathematical expectation of the source's information, or our *average decrease in uncertainty* when observing the symbols emitted by the source. If we use base 2 for the logarithm, entropy is of course measured in *bits*.

[6] A memory-less source is a source in which the probability of each symbol is independent of any previous symbols emitted by the source. This is not a very usual case. Typically, sources have memory and are called Markov sources of order *n*, where *n* is the number of previous symbols upon which the current symbol is dependent. A human English speaker is a Markov source of order ~32.

Entropy, as defined by Shannon[7], is a powerful function that accurately describes the mean information that can be emitted by a source. It has several, properties, some of them obvious, some of them not so intuitive:

(i) $H(X) \geq 0$

(ii) $H(X)=0$ *if and only if* $n=1$, $p=1$

(iii) $H(X) \leq \log(n)$

(iv) $H(X)=\log(n)$ *if and only if* $p_i=1/n$ *(equiprobability)*

Properties (i) and (ii) are relatively obvious. If a source is emitting something, it will always convey some information. The only case in which it will not do so is when the source always emits the same symbol, over and over again, with probability 1 (no surprise there, hence no information).

Property (iii) is known as the Fundamental Theorem of Entropy and can be derived by applying Gibbs inequality. Property (iii) and its corollary (iv) are somewhat counterintuitive to many students of Information Theory. Intuition might lead you to predict that a bias in the results of an experiment would be averaged out when computing entropy. The intuitive reasoning is more or less as follows (for a coin biased towards tails). A head yields more information (because it is rarer), but also happens less often. A tail gives less information (because it is the common result), but happens frequently. When you take the weighted average, both things should even out and you would get the same information (on average) as with an unbiased coin. Right?

Entropy and intuition

Wrong. A head might be very informative in a tail-biased coin, but it cannot yield too much information, no matter what. The mathematical reason for this effect is that information is a log function while probability is linear. This can be illustrated for the coin tossing problem (the formal treatment of the coin tossing case is known as binary entropy). In this case, we have only two possible outcomes (heads/tails) and we can define the probability of heads as *p* and the probability of tails simply as *1-p* (since probabilities should add to 1). In Figure 5 we can see that as the probability of heads *p* increases linearly, the information ($-\log(p)$) decreases logarithmically. So, looking at it the other way, we can

[7] A funny anecdote on the Shannon's adoption of the term *entropy* for his weighted mean of information can be found in WikiPedia's History of Entropy. Following (Jaynes Jaynes, E. T. (1957). "Information Theory and Statistical Mechanics." Physical Review **106**(4): 620.), one is inclined to think that von Neumann's comment was not so witty.

already see that, yes, as heads become less probable (tail-biased coin), we get more and more information. But when we compute entropy, we take into account *weighted* information (–p·log(p)). We can see that, in terms of weighted information, decreasing the probability of an outcome is a path of diminishing returns that peaks around p=0.3635. Afterwards, the increase in information (–log(p)) is relentlessly driven down by the decreased probability. Eventually, we get no *weighted* information because the *infinitely informative* outcome just *never* happens (p=0).

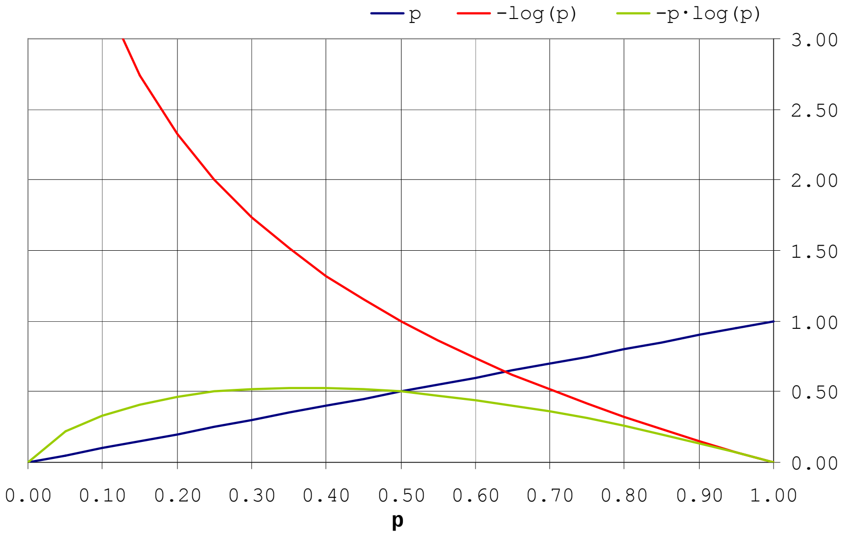

Figure 5 – Plot of probability (p), information (–log(p)) and its weighted measure (–p·log(p)) for different values of *p*.

This resolves part of our intuitive paradox: we have seen that making heads improbable is no guarantee that "head" information will rise proportionally. But, what about tails? Surely, having lots of tails, even if not very informative, would compensate the diminishing returns of weighted information from very infrequent heads? The answer is obviously "no", as it can already be appreciated in Figure 5. If tails become too frequent, their information (–log(p)) decreases logarithmically to zero, and their heightened probability (p→1) cannot do much to change the resulting weighted information (–p·log(p)). So, whenever we move away towards p=0 or p=1 (1–p=1 or 1–p=0), we end up with little precious information for our weighted mean (entropy). The only remaining question is: where is the optimal point? We have seen that p=0.3635 maximizes weighted information (–p·log(p)), but this is just one of the terms in our weighted mean (entropy). In the coin tossing experiment, we can plot both probabilities (p and 1–p), their weighted information and the resulting weighted mean (*H*). This gives us graphically Property (iv): entropy is maximized for equiprobable outcomes (p=0.5).

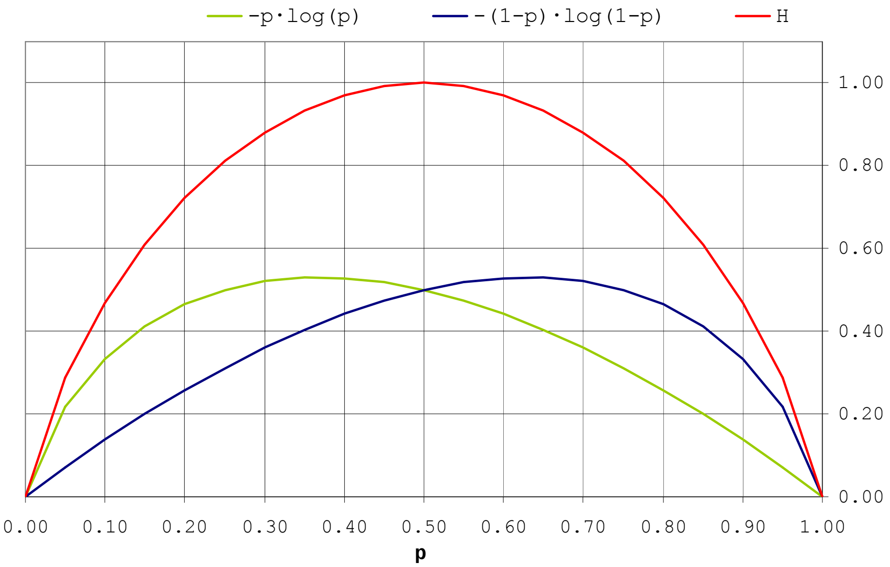

Figure 6 – Plot of entropy (*H*) and weighted information (`-p·log(p) / -(1-p)·log(1-p)`) for the two possible outcomes of an experiment and different values of *p*.

This result can be of course demonstrated mathematically for cases other than coin tossing (binary), but it is also rather intuitive if one thinks it through. Suppose we had a friend with whom we meet for lunch everyday. He always makes it. Of course, if one day he cannot make it for lunch, this will be an extraordinary event that will surprise us a lot (i.e. lots of information). But, on average, our friend does not surprise us much: he always shows up. On the contrary, if we meet another friend for dinner and she often has trouble keeping up with us, for a variety of unpredictable reasons, we are constantly in doubt: will she come for dinner tonight? Whether she shows up or calls to say she cannot make it, she is surprising us. If she keeps doing this with equal frequency, even though she might incur in short runs of, say, five successive days making it, on average she is surprising us a lot (maximally indeed).

Again, a word of caution, lest our blessed intuition misfire again. We humans tend to get bored or annoyed by apparently random events. We seek patterns. So it is likely that we might say that our friend is predictable (i.e. boring or most probably annoying) because she fails to show up as often as she makes it for dinner. Using perfect logic, we thereby conclude that she is not surprising us. Wrong! The probability of each result is indeed predictable (0.5), just as our lunch friend probability of making it (0.999) was predictable. But our dinner girlfriend surprises us each time, and maximally on average, while our lunch boyfriend just surprises us every so often, and marginally on average.

Entropy and entropy rates

As mentioned above, entropy measures the expectation of the information of a source. We can put this knowledge into practice by considering a source of information and analyzing its entropy. A typical example would be an English speaker or book. Assuming he/it is a memory-less source, we would get a result close to 4.14 bits per letter (Shannon 1950) based on the independent frequencies of each letter in English texts. Notice that now the units are specified as *bits per letter*, as they measure the entropy for any given character. This is sometimes confusing, as they seem to imply that we are measuring a *rate*, whereas we did not do so before. This is simply a matter of terminology, because we are now thinking of the speaker/book as someone/thing emitting a substantial number of symbols, not a single symbol, and we are thus interpreting the average information as a *rate* of information per symbol. Following this same reasoning, we could have said above that the die-roll entropy was in *bits per die roll*, but we never bothered. We assumed it to be so implicitly because we were thinking about the entropy (average information) of any *single* roll of dice. This is not to say that information cannot be measured in rates. Typically, information is transmitted serially through time. The number of bits transmitted each second is referred to as the *information rate* and is thus consequently measured in *bits per second*.

Crunching some entropy numbers

But let's try to move back to biology and apply our knowledge of entropy and information to something useful. Say, for instance, a genome. What is the entropy of a genome? Is there such a thing?

Clearly, the answers are yes and yes. A genome is most definitely a *source* of stored information and its entropy can be therefore computed. What symbols does a genome have? Typically four: A, C, T and G. What are their probabilities? Well, again, if we have the sequence of the whole genome, we can infer probability from the relative frequency of each letter (again, we consider here the genome to be a memory-less source, which is not a truthful approximation but simplifies things enormously (Phillips, Arnold et al. 1987; Pride, Meinersmann et al. 2003; Erill and O'Neill 2009)).

Let's analyze a real genome, and let's start with the archetypical one: *Escherichia coli*, our usually benign gut-pal. The relative base frequencies for the *E. coli* k-12 genome are:

```
%A    %T    %C    %G
24.6  24.6  25.4  25.4
```

So, we can now compute the information for each base:

$$I(A) = I(T) = -\log_2(0.246) = 2.023 \text{ bits}$$

$$I(G) = I(C) = -\log_2(0.254) = 1.977 \text{ bits}$$

Therefore, our entropy is:

$$\mathbf{H(Eco)=}\ 0.246\cdot2.023 + 0.246\cdot2.023 + 0.254\cdot1.977 + 0.254\cdot1.977 = \mathbf{1.999} \text{ bits}$$

This is the average information of any base in the *E. coli* genome and, as expected, it is close to 2 bits (which is the maximum information we can get from an experiment with 4 possible outcomes), because the base frequencies are almost equiprobable. Again, we can think of it as the rate of information in the *E. coli* genome, whatever suits us better. The neat result is that it is close to the maximum optimal value, which is what we are too often tempted to think about the result of an evolutionary process.

We can compute the entropy for another organism, however, and see that this is not always the case. Take another bacterial genome. For instance, that of the famous *Thermus aquaticus*, known around the world as the bacterium that provided the original Taq-polymerase enzyme used in the polymerase chain reaction (PCR). Let's get the base frequencies again[8]:

```
%A    %T    %C    %G
15.3  15.3  34.6  34.6
```

Our entropy now is:

$$\mathbf{H(Taq)=}\ 0.15\cdot2.73 + 0.15\cdot2.73 + 0.35\cdot1.51 + 0.35\cdot1.51 = \mathbf{1.876} \text{ bits}$$

As expected, our new entropy is lower, because of the strong bias in %GC content. This is an adaptation of *T. aquaticus* to high temperatures (as GC pairs have triple bonds, for double bonds in AT pairs, and

[8] These frequencies are computed on the whole genome unfinished sequence of Thermus thermophilus Y51MC23 and are thus approximate. The results, however, are unlikely to change much upon completion of the final draft. Close relatives, like Thermus thermophilus, which have finished genomes, show very similar frequencies.

thus strengthen the double helix against thermal noise). This adaptation is shared by many microorganisms living in hot environments, such as many Archaea and all known members of the bacterial *Thermus* genus. The decrease in entropy (`1.999 - 1.876 = 0.123`) might not seem huge (6.15%), but remember that this is a decrease of entropy *per character*. When we consider the size of a typical bacterial genome (4 Mbp), then such a decrease gains relevance. If the genomes of *E. coli* and *T. aquaticus* were of identical size (4 Mbp), it would mean that the *E. coli* genome would be able to encode 492,000 bits *more* of information than that of *T. thermophilus*. Which makes you wonder: if the ancestral microorganism that first arose on Earth was living in conditions similar to those of *T. aquaticus*, as some geological and phylogenetic analyses seem to imply, wouldn't it have chosen another set of bases/nucleotides that gave it maximal information encoding capacity in those conditions?

An intuitive grasp on genomic entropy and coding

A common way to understand intuitively what entropy means is to use the game of city guessing, based partly on Epimenides paradox. Epimenides made a statement: "All Cretans are liars", which became famous because he himself was a Cretan. The implicit self-reference creates a paradox that cannot be fully resolved. Was he lying, or telling the truth? This is a particular instance of the liar paradox, embodied by the self-referring sentence "This sentence is false", which again cannot be resolved satisfactorily[9]. Paradoxes aside, in the game of city guessing, we are lost and our aim is to know in which of (two possible) cities we are, knowing that all the inhabitants of city *A* tell the truth and all the inhabitants of city *B* are liars. This is not an irresolvable situation, but the question now is: how many questions do we need to ask?

We can apply the same kind of reasoning to the *T. aquaticus* genome. Let's imagine that a friend of us calls to tell us that a new genome position of the *T. aquaticus* genome has been sequenced. We are impatient to know what base it is, but our friend is a teaser and only allows us to make binary (yes/no) questions. How many questions do we need? How do we proceed?[10]

[9] See (Hofstadter, D. R. (1979). Gödel, Escher, Bach: an Eternal Golden Braid New York, NY, Basic Books.) for an in-depth and lively discussion of self-reference paradoxes in mathematics, music and painting, leading to the infamous Gödel's incompleteness theorems.

[10] If you are still wondering how many questions are required to solve the city problem, the answer is one, not two. The trick is to make a compound (AND) question that forces the unwary citizen to collapse the paradox for you. Many different combinations work. One such instance is: "If you were from the other city and I asked you if I am in city *A* right now, what would you answer?" Let's imagine you are in city *A*. A citizen from city *A*, knowing that *B* citizens are all liars, must answer NO (because that is the answer a B citizen would give). A *B* citizen, however, knows that an *A* citizen would tell you the truth (yes, you are in city *A*), but has to lie compulsively about it and answer NO. If you happen to be in city *B*, things get reversed. An *A* citizen knows that a *B* citizen would lie YES to you, and tells you so. Conversely, a *B* citizen

It is easy to see that the optimal questions (and associated answer probabilities) should be as follows:

```
Is it a G? → Yes (0.35)  → G
           → No  (0.65)  → Is it a C? → Yes (0.53)  → C
                                      → No  (0.47)  → Is it a T? → Yes (0.5)  → T
                                                                 → No  (0.5)  → A
```

Our first question obviously has $p=0.35$ of getting the right answer. This means than 35% of the time (if we were to replay the game), we would only need one question. When we need a second question, our probability will now be higher of getting it right (53%), and we will only need a third question on the remaining 47%. Thus, on average, we will need one question 35% of the time, two questions 34% of the time (that is 53% of the initial 65%) and three questions in the remaining 31% (47% of 65%). If we take the weighted mean, we get: $1\cdot0.35 + 2\cdot0.34 + 3\cdot0.31 = 1.96$ questions. Notice that this value is higher than the entropy we computed for the *H. aquaticus* genome (1.876 bits). This is because only our last question has an equiprobable answer. If we could approach the problem using only binary questions with equiprobable answers, we would get one bit of information per question (that is, in fact, the definition of *bit*). As it is, however, there is no way to formulate the questions here so that they yield equiprobable answers. Our first question, for instance, does not have equiprobable answers and, thus, it is not giving us all the information it could give us if it were equiprobable (it is, in fact, yielding $0.35\cdot1.51 + 0.65\cdot0.62 = 0.93$ bits). What does all this mean? It means that *entropy gives us the minimum possible average number of binary questions required to learn the outcome of an experiment*. This minimum would only be reached in the event that we could always pose binary questions with equiprobable outcomes.

Even though it may sound like a futile exercise, the guess-the-city game or its genome equivalent above lead to a very pointed question in computer science and communications theory. If you rephrase the problem, the number of binary questions we need to ask to determine a *T. aquaticus* base can be seen as the number of binary digits we need to encode a *T. aquaticus* base in binary form. We can easily see that we can only get away with one binary digit if we can pose an equiprobable question (this is the only case in which information theory and computer science *bits* are equivalent). Thus, the trivial question of how many questions we need becomes a very important question in coding theory. We

knows that *A* citizen would tell you the truth (no, you are not in city *A*) and has again to lie about it and answer YES. So, with one question, we know that in the ideal scenario, we could pose equiprobable questions and thus each binary digit would be conveying 1 bit of information. Whenever we move from that ideal situation, we are not exploiting each binary digit to carry one bit of information, so we need more binary digits (i.e. questions) to convey the same message. As we have seen, entropy tells us the minimum number of questions (i.e. binary digits) required, but that applies only to the ideal scenario (equiprobable questions). As you can imagine, compressing data is an important aspect of communications (the more compressed, the more you can send over the same line in the same time period), so figuring out how to encode information using the minimum number of bits is of essence in communications. This is equivalent of finding ways to ask questions that maximize their equiprobability and it is a fundamental problem, known as *source coding*, in coding theory (see Schneider's *Information Theory Primer* (Schneider 1995) for a worked out example of a Shannon–Fano coding as applied to genomic sequence).

On mutual information: communication

So far, we have been dealing exclusively with the *source* of a message (implicitly assuming that we are able to observe the emitted symbols), but we said earlier that Information Theory is concerned mainly about how information is *transmitted* from a *source* to a *receiver* by means of a *channel*. In an ideal world, our definition of entropy (*H*) above would suffice for this purpose. In an ideal world, we would have a noise-free channel. This is of course impossible in the real world, but we can make things robust enough to assume noise-free operation. Hence, if we roll a die and we observe the results (and we remain close to the die, and it is not foggy, and we don't have a severe vision impairment and…), the die entropy is basically the information we obtain in the communication process. So, in this case the die is the *source*, our visual path through air is the *channel* and we are the *receiver*. Although the channel is not noise-free, we would in our everyday life assume so.

Noise and conditional entropy

Let us go back now to the case of tossing a coin. Instead of tossing it on the floor and observing the coin toss directly, however, we now throw it out of our balcony on the 13th story of a Las Vegas hotel. Fortunately for us, a friend of us is down there to take a look at the coin and tell us the result. But, let us stretch the imagination and suppose this is a pre-cell phone era (yeah, way, way back then) and that our friend has to shout the result back to us. Now our channel is not so nice anymore. Air is what it is:

know that NO means we are in city *A* and that YES means we are in city *B*.

there is always some wind, other sounds interfering, sound decays with distance from the sound source following an inverse square law, and more so the drier the air is… So, yes, we have noise. And to proceed with the argument, let us assume that we mishear "head" for "tail" (and vice versa) one in every 100 (1%) coin tosses.

If we are using a fair coin, then our source entropy is clearly $H(X)=-0.5\cdot\log_2(0.5)\cdot 2= 1$ bit. We can factor in noise now, though, and compute the entropy *after the coin has been tossed and we have heard the shouted result*. This is called *conditional entropy* and, mathematically, it can be expressed as:

$$H(X|Y)= -[0.99\cdot\log_2(0.99) + 0.01\cdot\log_2(0.01)]= -[-0.014 - 0.066]= 0.081 \text{ bits}$$

Conditional entropy is also very intuitive to grasp. It expresses our uncertainty (as receivers) on the result of an experiment *after the experiment has been carried out*. In the case of the friend-relayed coin toss, *X* is the result of the experiment and *Y* is what we hear (or mishear) as being the result. *X|Y* is the conventional notation in Bayesian probability to denote *X* once we know *Y*. That is, conditional probability [probability of *X* once *Y* is known] and, therefore, conditional entropy. Thus, the equation above can be interpreted as follows. We toss the coin and the outcome is *X*. Our friend shouts and we hear *Y*. Knowing *Y* (the result of the experiment as we perceive it), *H(X|Y)* expresses our remaining uncertainty over *X* (the *real* result of the experiment). Since we mishear 1% of the shouted results, and knowing that *X* is in an already fixed state (either heads or tails), from our receiver standpoint the two possible outcomes of the experiment are now *X=Y* and *X≠Y*, with probabilities $P(X=Y|Y)=0.99$ and $P(X\neq Y|Y)=0.01$. This leads to $H(X|Y)=0.081$ bits. In other words, the second "experiment" of having our friend shout the results is not very "informative" because it is extremely biased: we are not very uncertain about the *real* coin toss result due to our friend shouting the result.

Mutual information

Of course, our second "experiment" was no experiment at all. Our friend (and his shouting and the air and so on) were the *channel* of a communication system consisting of a *source* (the coin), a *receiver* (us) and a *channel* (everything in between). And, of course again, we really don't want this second "experiment" to be "informative" at all. Ideally, we would like a noise-free channel; that is, a channel in which $P(X=Y|Y)=1$ and thus $H(X|Y)=0$. In this ideal world, we would be completely certain about the outcome of the coin toss after hearing the result. The complete opposite would be a noise-ridden channel, in which we misheard our friend with probability 0.5. In this nightmarish world, we would have no clue about the outcome of the coin toss even after hearing the result a thousand times.

What we are saying, effectively, is that *H(X|Y)* represents our uncertainty over *X* once we know *Y*, and thus it is intimately linked to the channel over which the original “message” known as *X* travels in order to arrive to us known as *Y*. Furthermore, we are also implying that *H(X|Y)* is a measure of *information loss* taking place in the channel or, equivalently, an *increase in uncertainty* due to transmission through the channel. This intuitively leads us to the concept of mutual information, as postulated originally by Shannon (Shannon 1948):

$$I(X,Y)=H(X)-H(X|Y) \qquad (8)$$

This definition of information (*I*) takes now explicitly into account that there is a source emitting *X*, a receiver receiving *Y* and a channel with conditional probabilities *P(X|Y)* for every pair of *X* and *Y* values. More formally, it is defined assuming a source ξ emitting symbols from an alphabet $X=[x_1,...,x_n]$, with associated probabilities $P_X=[p_{x1},...,p_{xn})$, a receiver ψ receiving symbols from an alphabet $Y=[y_1,...,y_n]$ with probabilities $P_Y=[p_{y1},...,p_{ym})$ and a transmission channel defining a matrix of conditional probabilities $P_{A|B}(i,j)$ that contains the probability that the received symbol y_j corresponds to the emitted symbol x_i, for every possible value of i and j.

By following the definition of *I* we can already see how *H(X|Y)* is treated as information loss (or increase in uncertainty). In any communication process, our goal is to transfer information and, consequently, decrease the uncertainty of the receiver by conveying the information to him. As a measure of entropy, *H(X)* measures the amount/rate of information the source can emit. *H(X|Y)*, on the other hand, measures the amount/rate of information that is lost to noise (i.e. the increase in uncertainty introduced by the channel). The difference, $I(X,Y)=H(X)-H(X|Y)$ is the amount/rate of information from the source that can be effectively received by the receiver through the channel. In other words, *I* is the average *decrease in uncertainty* the receiver experiences during the communication process. This decrease in uncertainty is provided by knowledge of the result of the experiment (X) and hindered by noise in the process of obtaining that knowledge (X|Y). Mutual information has many interesting properties, like being non-negative and symmetrical. It also agrees with our intuitive notion of information transmission for trivial cases. Given a noise-free channel, $H(X|Y)=0$ and then $I=H(X)$. That is, we are able to convey all the information in *X* to the receiver, which leads us back to the original Hartley/Shannon setup in which information is exclusively determined by our *a priori* uncertainty (because our *a posteriori* uncertainty is zero). Given a noise-ridden channel, *X* and *Y* become independent, so that $H(X|Y)=H(X)$ and $I=0$. No information is transmitted; there is no decrease in entropy for the receiver.

Information by example

Shannon illustrated beautifully the concept of mutual information and its pitfalls with a simple example (Shannon 1948). Imagine a source emitting 2 equiprobable symbols at a rate of 1000 symbols per second. Obviously, the source rate is 1000 bits/s following the definition of *bit*. We imagine now that there is noise in the channel and that one out of one hundred symbols (1%) are received incorrectly. Shannon then puts forward the question of the rate of information *transmission*. It is obvious that this must be lower than 1000 bits/s, since there are errors due to noise. But how much lower? Shannon first gives the intuitive (yet wrong) answer: the information rate is 990 bits/s because 1% (10 our of 1000) of the emitted symbols are received incorrectly (1000 – 10= 990). He then proceeds to demolish this argument, on the basis that the receiver does not know *when* the errors take place, and this further limits the information rate. Clearly, if the noise level was as large as to make received symbols independent from the emitted ones, every received symbol would have a probability of 0.5 of being one of the two possible emitted symbols. This means that 50% of the symbols would arrive correctly just by chance. Following the intuitive argument, we would thus be crediting the channel as being able to transmit at a rate of 500 bits/s, when in fact emitted and received symbols would be completely independent. Having destroyed the intuitive answer, Shannon proceeds to show that the proper correction for the ideal capacity of the channel (1000 bits/s) is the amount of information lost to noise in the channel. That is, namely, *H(X|Y)* as defined by the noise in the channel and the source *H(X)*. Shannon defines the channel capacity as the *maximum* mutual information *I(X,Y)*. Obviously, there are two very different ways of approaching an optimum channel capacity. Physically, electrical engineers can try to improve the channel capacity by tweaking directly with the physical properties of the channel to reduce the impact of noise and thus minimize *H(X|Y)*. Mathematically, coding theorists take a given noise in the channel for granted and instead focus their efforts in obtaining an encoding that maximizes the source entropy *H(X)* while minimizing the conditional entropy *H(X|Y)*. In a noiseless channel, this translates simply in obtaining an encoding that maximizes the source entropy *H(X)*. In a channel with noise, however, maximizing *I(X,Y)* to reach the channel capacity becomes a complex optimization process because encodings that favor large *H(X)* will typically also yield large *H(X|Y)*[11].

Information in transcription factor-binding motifs

After this long detour, we can come back to our original question: why do bits appear in the X-axis of our BUH TF-binding motif information logo?

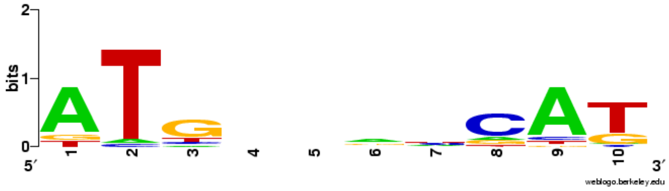

Figure 7 – Sequence logo for the BUH binding motif, as shown originally in **Figure 3**.

The long due answer is quite simple once you grasp the original insight of Schneider and coworkers (Schneider, Stormo et al. 1986), based on previous work by Gatlin (Gatlin 1966; Gatlin 1968), that the binding of a TF to its DNA sites is, in essence, an information process. In order to see this, imagine a stretch of 10 bases anywhere in the *E. coli* genome. What is your uncertainty about the base occupying each position of this *site*? Remember that we can envisage a genome as a source of information and, therefore, we can compute its entropy. In this case, given an alphabet Ω of four bases (A, C, T and G), we obtain:

$$H(X) = H_{before}(l) = -\sum_{S\in\Omega}\left[f(S)\cdot(\log_2\left(f(S)\right)\right] \tag{9}$$

This is our *a priori* the entropy (our average uncertainty), *before* any information transmission (mediated by proteins or by any other means) has taken place. It is, as defined above, the entropy of the *source*. For the *E. coli* genome, we know from previous computation that this value is approximately 1.999 bits. In other words, we cannot guess much about any position of the sequence and do better than random chance.

[11] Intuitively, a simple way to overcome the effect of noise in the channel is to introduce redundancy in the message (e.g. using two symbols instead of one), but this will have the net effect of reducing the source entropy and lowering I(X,Y).

Now let us consider what happens if we are made aware (by whatever experimental means), that protein BUH binds the site we were thinking about. Do things change now? What is our uncertainty over each of the 10 positions of the site *once we know* that protein BUH binds there? As we saw previously, BUH does not bind always the same identical sequence, but rather a group (or collection) of similar sequences. We can recall that from this collection we were able to compute a position specific frequency matrix (PSFM):

	1	2	3	4	5	6	7	8	9	10
A	0.76	0.04	0.08	0.28	0.12	0.44	0.24	0.12	0.80	0.04
C	0.00	0.04	0.12	0.32	0.28	0.12	0.28	0.68	0.08	0.04
T	0.12	0.92	0.16	0.16	0.28	0.12	0.40	0.08	0.08	0.68
G	0.12	0.00	0.64	0.24	0.32	0.32	0.08	0.12	0.04	0.24

Table 5 – Position Specific Frequency Matrix for transcription factor BUH, as shown originally in **Table 3**. Each position in the matrix defines the probability *p(S_l)* that position *l* is occupied by base *S* in the BUH collection.

Seen in the light of Information Theory, the PSFM is now telling us something new. It is giving us the conditional probabilities P(X|Y). That is, it is telling us the probability that base *S* occupies position *l* in our randomly picked site, *once we know* that the protein binds there. It is therefore not very difficult to compute the conditional entropy *H(X|Y)* for each position based on the PSFM. That is, our remaining uncertainty over which base occupies each position *after* we know that the protein binds the site:

$$H(X\mid Y)=H_{after}(l)=-\sum_{S_l\in\Omega}\left(p(S_l)\cdot\log_2\left(p(S_l)\right)\right) \qquad (10)$$

In other words, recognition of a site by a protein is a noisy process and the information *loss* due to this noise is represented by *H(X|Y)*.[12] The protein is thus acting as the *channel* in our communication

[12] When we first introduced mutual information, we defined source ξ and receiver ψ with their respective alphabets X and Y. For simplicity, in the coin tossing example that ensued, we assumed that X and Y were the same (heads and tails). However, this does not need to be the case, and it is clearly not the case in the protein recognition setting. A better analogy for the protein recognition problem would be rolling a die (X=[1,2,3,4,5,6]) and having a friend waving (or not) at us with a flag (Y=[yes/no]). Our friend could wag the flag whenever the result were greater than 3 (4,5,6) and not wave otherwise. In this case, our friend and his flag wagging (assuming no other factors intervened: he can observe the wagging directly, with no interference) would act as the communication channel. If the die is unbiased, the probabilities are $p(X=x_i)=0.166$ for each outcome, leading to an original entropy of H(X)=2.58 bits. By seeing the flag wag, our original uncertainty over the result of the unbiased die roll would be reduced, but the transmission process would be very noisy (due to the dimensionality reduction induced by the alphabet change). Assuming our friend had no biases for any number when flagging, if we saw the flag being wagged our new estimate on the probabilities would be $p(X=x_i|Y)=0.33$ for 4, 5 and 6 and $p(X=x_i|Y)=0$ for 1, 2 and 3. Our uncertainty after seeing the flag wagging would then be H(X|Y)=1.58 bits and, thus, our mutual information would be I(X,Y)=1 bit. So, we would have gotten information from our friend on the outcome of the experiment (1 bit), but we would still be uncertain over the actual result. Same goes for a protein. By binding it is giving us information on an otherwise unknown sequence, but since binding is not 100% specific, we are left with some uncertainty over the real base at each position.

process (we assume here that our observing the binding of the protein to the site is a noise-free process[13]). Mutual information for each site position can be expressed as:

$$I(l) = H_{before}(l) - H_{after}(l) = \left[-\sum_{S\in\Omega} \left(f(S) \cdot (\log_2 \left(f(S) \right) \right) \right] - \left[-\sum_{S_l\in\Omega} \left(p(S_l) \cdot \log_2 \left(p(S_l) \right) \right) \right] \qquad (11)$$

Mutual information *I(l)* was labeled as $R_{sequence}(l)$ by Schneider and coworkers when it was first applied to the field of molecular biology (Schneider, Stormo et al. 1986). Assuming positional independency, $R_{sequence}(l)$ can be added for all the positions in the motif and the overall $R_{sequence}$ measure is often referred to as the *information content* of the binding motif.

We can now revisit our BUH logo:

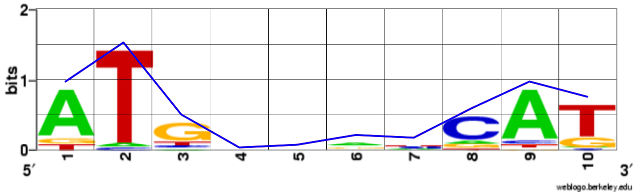

Figure 8 – Sequence logo for the binding motif BUH from **Figure 3**. The $R_{sequence}$ function is superimposed on the logo.

We can see now that the sequence logo is a clever combination of frequency information (Figure 4) and information content. Letter height is an indicator of base frequency per position, while the letter-stack height is an indicator of the information content per position ($R_{sequence}(l)$). In case you are wondering, the overall information content value for BUH ($R_{sequence}$) is 5.83 bits.

Again, we can apply intuitive reasoning to trivial cases. A transcription factor targeting only one specific sequence, with no variation whatsoever, is equivalent to a noise-free channel. The PSFM will be made in this case of columns filled with a one and three zeros. Whenever we know that the protein is bound, we will have no doubt at all about the base occupying each position. The logo would look like:

[13] This is of course false, but nevertheless a valid assumption, in the same league (but not scale) as the assumption of our ability of directly observing without noise the result of a die roll by visual inspection. Binding sites for proteins are inferred by experimental assays that are typically carried out in triplicate (at least), as a way to minimize the noise inherent to the experimental procedure. Different experimental procedures (and the controls used therein) will provide different degrees of certainty with regards to protein binding to a specific site.

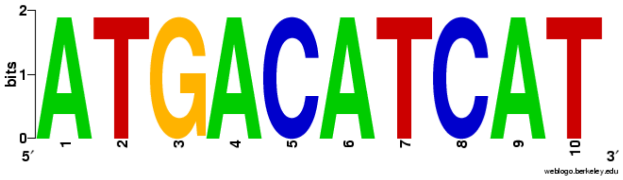

Figure 9 – Sequence logo for an omnipotent TF that recognizes only one specific kind of sequence (`ATGACATCAT`), with no allowed variants.

Mutual information is maximal in this case (for a given genomic entropy *H(X)*), which means that our average decrease in uncertainty over the bases occupying each position of the sequence once we know the protein binds there is also maximum and equal to *H(X)*.

The opposite case is a completely unspecific TF, which binds anywhere in the genome. In this case, the PSFM will be filled with ~0.25 values (the exact numbers will depend on the background distribution). In other words, by knowing that the protein binds there we will gain no information on what bases occupy each position of the motif. Mutual information, that is, our decrease in uncertainty after observing binding, will thus be minimal (i.e. zero).

Another take on mutual information

When thinking about the sequences constituting a binding site collection, we should not forget that these (for real TFs unlike BUH) are real sequences and that they are shown neatly aligned in the collection. We can interpret mutual information as the reduction in uncertainty we experience with regard to a random genome sequence whenever we are told that a particular TF binds there. However, since the sequences in our TF collection are real genomic sequences from a real genome, we can also interpret mutual information as the reduction in entropy experienced by certain genomic stretches during their evolution from “random” genomic sequence into TF-binding sites. Mutual information can thus be seen as a measure of conservation (or redundancy) in the positions of the TF-binding motif. Since maintaining the redundancy (or conservation) of these positions against random mutation is an energy-consuming process (i.e. natural selection has to weed out those individuals in whom a number of positions of a number of TF-binding sites are mutated), it is logical to assume that highly redundant/conserved positions are more important to the binding process than less conserved/redundant positions. This line of reasoning has been advocated by different authors and the term *Redundancy Index* has been proposed as an alternative name for mutual information (a.k.a. *information content*) in the context of TF-binding motifs ($R_{sequence}$) (O'Neill 1989).

Acknowledgements

I would like to thank Tom Schneider and Mike O’Neill for taking the time to read earlier versions of this manuscript and for providing invaluable insights and amendments. Thanks also to Gary Stormo for insightful comments and to Kenneth Smith for pointing out several errors.

References

Ben-Naim, A. (2007). Entropy demystified : the second law reduced to plain common sense. Hackensack, N.J., World Scientific.

Collado-Vides, J., B. Magasanik, et al. (1991). "Control site location and transcriptional regulation in *Escherichia coli*." Microbiol Rev **55**(3): 371-94.

Crooks, G. E., G. Hon, et al. (2004). "WebLogo: a sequence logo generator." Genome Res **14**(6): 1188-90.

Erill, I. and M. C. O'Neill (2009). "A reexamination of information theory-based methods for DNA-binding site identification." BMC Bioinformatics **10**(1): 57.

Gatlin, L. L. (1966). "The information content of DNA." J Theor Biol **10**(2): 281-300.

Gatlin, L. L. (1968). "The information content of DNA. II." J Theor Biol **18**(2): 181-94.

Hartley, R. V. L. (1928). "Transmission of Information." Bell System Technical Journal **7**: 535-543.

Hofstadter, D. R. (1979). Gödel, Escher, Bach: an Eternal Golden Braid New York, NY, Basic Books.

Jaynes, E. T. (1957). "Information Theory and Statistical Mechanics." Physical Review **106**(4): 620.

Marks, J. (2002). What it means to be 98% chimpanzee : apes, people, and their genes. Berkeley, CA ; London, University of California Press.

O'Neill, M. C. (1989). "Consensus methods for finding and ranking DNA binding sites. Application to *Escherichia coli* promoters." J Mol Biol **207**(2): 301-10.

Phillips, G. J., J. Arnold, et al. (1987). "Mono- through hexanucleotide composition of the *Escherichia coli* genome: a Markov chain analysis." Nucleic Acids Res **15**(6): 2611-26.

Pride, D. T., R. J. Meinersmann, et al. (2003). "Evolutionary Implications of Microbial Genome Tetranucleotide Frequency Biases." Genome Res **13**(2): 145-158.

Robbins, R. J. (1992). "Challenges in the human genome project." Engineering in Medicine and Biology Magazine, IEEE **11**(1): 25-34.

Schneider, T. D. (1995). Information Theory Primer.

Schneider, T. D. (2002). "Consensus sequence Zen." Appl Bioinformatics **1**(3): 111-9.

Schneider, T. D. and R. M. Stephens (1990). "Sequence logos: a new way to display consensus sequences." Nucleic Acids Res **18**(20): 6097-100.

Schneider, T. D., G. D. Stormo, et al. (1986). "Information content of binding sites on nucleotide sequences." J Mol Biol **188**(3): 415-31.

Shannon, C. E. (1948). "A mathematical theory of communication." Bell System Technical Journal **27**: 379-423 623-656.

Shannon, C. E. (1950). "Prediction and Entropy of Printed English." Bell System Technical Journal **3**: 50-64.

Tillman, F. and B. Roswell Russell (1961). "Information and entropy." Synthese **13**(3): 233-241.

Varela, F. G., H. R. Maturana, et al. (1974). "Autopoiesis: the organization of living systems, its characterization and a model." Curr Mod Biol **5**(4): 187-96.