Information Theory Primer

With an Appendix on Logarithms

Postscript version: ftp://ftp.ncifcrf.gov/pub/delila/primer.ps
web versions: http://www.lecb.ncifcrf.gov/∼toms/paper/primer/

Thomas D. Schneider*

version = 2.59 of primer.tex 2007 Jan 28

Diese Einführung wurde für Molekularbiologen geschrieben, die mit der Informationstheorie normalerweise nicht vertraut sind. Das Ziel des Textes besteht darin, eine Einführung in die grundlegenden Ideen zur Informationstheorie zu geben, damit weiterführende Literatur verstanden werden kann [1, 2, 3, 4, 5, 6, 7, 8, 9]. Das meiste von dem, was hier steht, kann auch in anderen Einführungen zur Informationstheorie gefunden werden. Obwohl Shannons Arbeit über die Informationstheorie im Original [10] teilweise schwer zu verstehen ist, ist sie an anderen Stellen gut verständlich. Überspringt die schwer verständlichen Teile und Ihr werdet die Arbeit genießen. Nach Shannons Arbeit veröffentlichte Pierce ein sehr populäres Buch [11], das eine großartige Einführung zur Informationstheorie ist. Weitere Einführungen sind in der Literturangabe [1] angeführt. Ein hilfreiches Arbeitsbuch könnte die Literaturangabe [12] sein. Die gesammelten Werke Shannons sind in [13] publiziert worden. Hinweise, wie das Buch bestellt werden kann sind unter

> http://www.lecb.ncifcrf.gov/∼toms/bionet.info-theory.faq.html
> #REFERENCES-Information_Theory

zu finden.
Weitere Texte und Programmdokumentation können unter

> http://www.lecb.ncifcrf.gov/∼toms/

gefunden werden.

Beachte: Falls ihr an manchen Stellen auf Unklarheiten oder Probleme in dieser Einführung treffen solltet, dann schreibt mir bitte eine E-Mail, und beschreibt das Problem und die genaue Textstelle. Sind die Hinweise gerechtfertigt, dann wird der Text an der betreffenden Stelle geändert, um die Einführung so verständlich wie möglich zu gestalten. Ich danke allen Leuten, die Fehler ausgemerzt haben und damit die vorliegende Version ermöglichten.

---

*National Cancer Institute, National Cancer Institute, Laboratory of Experimental and Computational Biology, P. O. Box B, Building 144, Room 469, Frederick, MD 21702. email address: toms@ncifcrf.gov. http://www.lecb.ncifcrf.gov/∼toms/ Der Text stammt aus Kapitel II meiner Dissertation: “The Information Content of Binding Sites on Nucleotide Sequences”, University of Colorado, 1984. Da es sich um eine Arbeit für die U. S. Regierung handelt, kann das Dokument nicht kopiergeschützt werden. [Deutsche Übersetzung basiert auf Version 2.59, übersetzt von Fabio Tosques.]

Information und Unbestimmtheit

Information und Unbestimmtheit sind technische Begriffe, die einen Vorgang beschreiben, mit dem ein oder mehrere Objekte aus einer Sammlung von Objekten ausgewählt werden. Wir wollen uns vorerst weder um die Bedeutung noch um die Auswirkungen von Information kümmern, bevor wir nicht wissen, wie wir dies mathematisch tun können. Stellen wir uns vor, wir hätten ein Gerät, dass 3 Symbole ausgibt, A, B oder C. Während wir auf das nächste Symbol warten, sind wir *unsicher* (*unbestimmt*), welches Symbol erzeugt wird. Sobald ein Symbol erscheint und wir es sehen, sinkt die Unbestimmtheit und wir merken, dass wir etwas *Information* erhalten haben. Das bedeutet, dass durch die Information die Unbestimmtheit sinkt. Und wie messen wir die *Unbestimmtheit*? Der einfachste Weg wäre zu sagen, dass wir eine „Unbestimmtheit von 3 Symbolen" haben. Das funktioniert solange gut, bis wir anfangen, zur gleichen Zeit ein zweites Gerät zu beobachten, welches, sagen wir, die Symbole 1 und 2 erzeugt. Das zweite Gerät erzeugt also eine „Unbestimmtheit von 2 Symbolen". Verbinden wir die beiden Geräte zu einem Gerät, erhalten wir sechs Möglichkeiten, A1, A2, B1, B2, C1, C2. Dieses Gerät hat eine „Unbestimmtheit von 6 Symbolen". Auf diese Weise denken wir normalerweise nicht über Information nach, z. B. wenn wir zwei Bücher erhalten, sagen wir, dass wir doppelt soviel Information haben, wie von einem einzigen Buch. Das heisst, wir möchten,d dass unser Maß additiv ist.

Das können wir einfach erreichen indem wir den Logarithmus von der Anzahl der möglichen Symbole nehmen. Dann können wir die Logarithmen addieren und müssen nicht die Anzahl der Symbole multipluzieren. In unserem Beispiel erzeugt das erste Gerät eine Unbestimmtheit von $log(3)$, das zweite von $log(2)$ und die verbundenen Geräte von $log(3)+log(2) = log(6)$. Die Basis der Logarithmen bestimmt dann die Einheiten. Nehmen wir die Basis 2, dann sind die Einheiten in Bits (mit der Basis 10 hätten wir Digits und mit der Basis des natürlichen Logarithmus, e, hätten wir Nats [14] oder Nits [15]). Wenn also ein Gerät ein Symbol erzeugt, so haben wir eine Unbestimmtheit von $\log_2 1 = 0$ Bits, und wir besitzen keine Unbestimmtheit darüber, was das Gerät als nächstes erzeugt. Erzeugt das Gerät zwei Symbole, haben wir eine Unbestimmheit von $\log_2 2 = 1$ Bit. Beim Lesen einer mRNA, wo beim Ribosom jede der 4 Basen gleichwahrscheinlich ist, besitzen wir eine Unbestimmtheit von 2 Bits.

Wir stellen fest, dass unsere Formel für Unbestimmtheit allgemein $\log_2(M)$ lautet, mit M für die Anzahl der Symbole. Im nächsten Schritt wollen wir die Formel so erweitern, dass sie auch die Fälle lösen kann, wo die Symbole nicht gleichwahrscheinlich auftreten. Haben wir beispielsweise 3 Symbole, von denen eines nie auftritt, so haben wir wieder eine Unbestimmtheit von 1 Bit. Falls das dritte Symbol im Vergleich zu den anderen beiden nur sehr selten auftritt, dann müsste die Unbestimmtheit größer sein als 1 aber kleiner als $\log_2(3)$ Bits. Stellen wir die Formel etwas um:

$$\begin{aligned} \log_2(M) &= -\log_2(M^{-1}) \\ &= -\log_2(\frac{1}{M}) \\ &= -\log_2(P) \end{aligned} \tag{1}$$

mit $P = 1/M$ für die Wahrscheinlichkeit mit der ein Symbol auftritt. (Falls wir uns nicht an den Trick erinnern, wie man das „Zeichen herauslöst", bedenke dass $\log M^b = b \log M$ mit $b = -1$.)

Wir wollen dies nun für verschiedene Wahrscheinlichkeiten des Symbols P_i verallgemeinern, so dass die Summe der Wahrscheinlichkeiten gleich 1 ist:

$$\sum_{i=1}^{M} P_i = 1. \tag{2}$$

(Erinnern wir uns, dass das $\sum$ Zeichen bedeutet, die P_i zu addieren, für alle i beginnend von 1 bis M.) Die Überraschung, die wir erhalten, wenn das i^{te} Symbol erscheint wurde von Tribus [16] „surprisal" genannt und ist analog definiert durch $-\log_2 P$, also:[1]

$$u_i = -\log_2(P_i). \tag{3}$$

Ein Beispiel: wenn P_i sich der 0 nähert, dann sind wir *sehr überrascht*, wenn wir das i^{te} Symbol sehen (da es ja so gut wie nie erscheinen sollte), und die Formel ergibt, dass sich der Wert von u_i ∞ nähert. Andererseits sind wir, wenn $P_i = 1$ ist, überhaupt nicht überrascht, wenn das i^{te} Symbol erscheint (da es ja immer erscheinen soll) und daher ist $u_i = 0$.

Unbestimmtheit ist der *durchschnittliche Überraschungswert* für die unendliche Zeichenfolge von Symbolen, die unser Gerät erzeugt. Wir wollen nun versuchen den Durchschnitt einer Zeichenfolge der Länge N zu finden, deren Alphabet aus M Symbolen besteht. Nehmen wir an, dass das i^{te} Symbol genau N_i-mal erscheint, so dass

$$N = \sum_{i=1}^{M} N_i. \tag{4}$$

Dann erhalten wir N_i Fälle mit dem Überraschungswert u_i. Der durchschnittliche Überraschungswert für N Symbole ist:

$$\frac{\sum_{i=1}^{M} N_i u_i}{\sum_{i=1}^{M} N_i}. \tag{5}$$

Ersetzen wir den Nenner mit N und fügen N in der Summe im Zähler ein, so erhalten wir:

$$\sum_{i=1}^{M} \frac{N_i}{N} u_i \tag{6}$$

Wenn wir das für eine unendliche Zeichenkette von Symbolen durchführen, dann wird die Häufigkeit N_i/N zu P_i, der Wahrscheinlichkeit des i^{ten} Symbols. Führen wir diese Ersetzung durch, dann sehen wir, dass unser durchschnittlicher Überraschungswert (H) die folgende Gleichung erhält:[2]

$$H = \sum_{i=1}^{M} P_i u_i. \tag{7}$$

[1] Warum verwendet er 'u' als Symbol für die Überraschung? S ist das Standardsymbol für die Entropie, was nicht deutlich vom Zeichen s unterschieden werden kann, daher steht 'u' für Unsicherheit („uncertainty") ... uns gehen langsam die Symbole in molekularen Informationstheorie aus!

[2] Das Symbol H übernahm Shannon von Boltzmanns H-Theorem aus der Thermodynamik. Wir wir dem American Journal of Physics entnehmen können, benutzte Boltzmann anfangs das Symbol E und erst später das Symbol H, welches vermutlich vom griechischen Buchstaben η = Eta stammt [17, 18]. Das griechische Eta entspricht im Englischen dem Buchstaben H. Boltzmann wählte das H wahrscheinlich um Verwirrungen mit dem Symbol E zu vermeiden, das für Energie verwendet wurde.

Ersetzen wir dann noch u_i, dann erhalten wir Shannons berühmte Formel für Unbestimmtheit:

$$\boxed{H = -\sum_{i=1}^{M} P_i \log_2 P_i \quad \text{(Bits pro Symbol).}} \tag{8}$$

Shannon hat diese Formel auf einem weitaus rigoroseren Weg erhalten als wir, indem er verschiedene geeignete Eigenschaften für die Unbestimmtheit festlegte und daraus seine Funktion ableitete. Hoffentlich gibt der Weg, den wir genommen haben, Euch ein Gefühl dafür wie die Formel funktioniert.

Um zu begreifen wie die H Funktion aussieht, können wir sie für den Fall von zwei Symbolen zeichnen. Dies zeigt die Graphik Abbildung 1:[3]

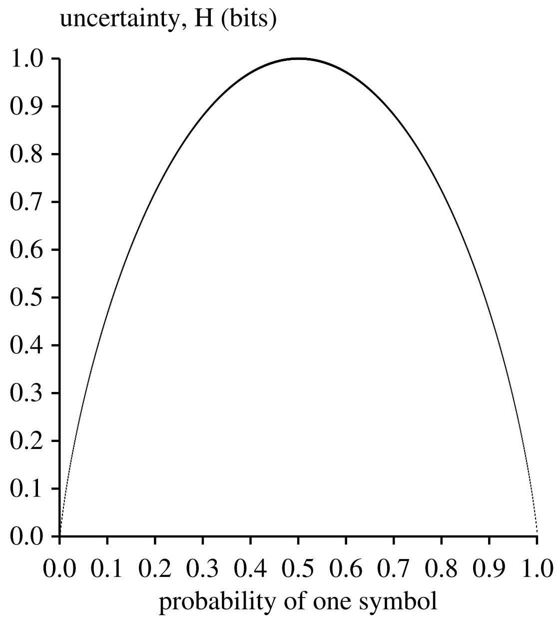

Abbildung 1: H-Funktion für zwei Symbole.

Beachte, dass die Kurve symmetrisch ist und sie das den höchsten Punkt erreicht, wenn die beiden Symbole gleichwahrscheinlich sind (Wahrscheinlichkeit = 0.5). Sie nähert sich der 0 sobald eines der Symbole auf Kosten des anderen immer größer wird.

Als lehrreiche Übung wollen wir nun annehmen, dass alle Symbole gleichwahrscheinlich sind. Wie lässt sich dann die Gleichung für H (Formel (8)) reduzieren?

****

[3] Das Programm, welches die Graphik erzeugt findet sich unter http://www.lecb.ncifcrf.gov/~toms/delila/hgraph.html

Gleichwahrscheinlich bedeutet, dass $P_i = 1/M$. Ersetzen wir das in der Gleichung für die Unbestimmtheit, so erhalten wir:

$$H_{gleichwahrscheinlich} = -\sum_{i=1}^{M} \frac{1}{M} \log_2 \frac{1}{M} \tag{9}$$

Da M keine Funktion von i ist, können wir M aus der Summe ausschließen:

$$\begin{aligned} H_{gleichwahrscheinlich} &= -(\frac{1}{M} \log_2 \frac{1}{M}) \sum_{i=1}^{M} 1 & (10) \\ &= -\left(\frac{1}{M} \log_2 \frac{1}{M}\right) M \\ &= -\log_2 \frac{1}{M} & (11) \\ &= \log_2 M \end{aligned}$$

Wir erhalten dieselbe einfache Gleichung mit der wir begonnen haben. Für eine gegebene Anzahl von Symbolen (d. h. M ist gegeben) nimmt die Unbestimmtheit H den größten Wert an, wenn alle Symbole gleichwahrscheinlich sind. Demnach ist es beispielsweise schwerer bei einer nicht manipulierten Münze das Ergebnis vorherzusagen als bei einer manipulierten. Eine weitere Übung: wie groß ist die Unbestimmtheit, wenn wir 10 Symbole haben und nur ein einziges von ihnen erscheint? (Hinweise sind: $\lim_{p\to 0} p \log \ p = 0$ mit $p = 1/M$ und die Regel von l'Hôpital, wo $0 \log_2 0 = 0$.)

Was bedeutet es nun wenn wir sagen ein Signal habe 1.75 Bits pro Symbol? Das bedeutet, wir können das ursprüngliche Signal in eine Zeichenfolge von Einsen und Nullen konvertieren (Binärziffern), so dass wir *im Durchschnitt* 1.75 Bits für jedes Symbol in der Originalnachricht benötigen. Einige Symbole brauchen mehr Bits (die selten auftretenden) und andere brauchen weniger Bits (die häufig auftretenden). Dazu ein Beispiel. Nehmen wir an, wir haben $M = 4$ Symbole:

$$A \quad C \quad G \quad T \tag{12}$$

mit den folgenden Wahrscheinlichkeiten (P_i):

$$P_A = \frac{1}{2}, \quad P_C = \frac{1}{4}, \quad P_G = \frac{1}{8}, \quad P_T = \frac{1}{8}, \tag{13}$$

welche im einzelnen den folgenden Überraschungswert haben ($-\log_2 P_i$):

$$u_A = 1 \text{ Bit}, \quad u_C = 2 \text{ Bits}, \quad u_G = 3 \text{ Bits}, \quad u_T = 3 \text{ Bits}, \tag{14}$$

dann ist die Unbestimmtheit

$$H = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = 1.75 \text{ (Bits pro Symbol)}. \tag{15}$$

Wir wollen dies nun so umkodieren, dass die Anzahl der Binärziffern dem des Überraschungswertes entspricht

$$\begin{aligned} A &= 1 \\ C &= 01 \\ G &= 000 \\ T &= 001 \end{aligned} \tag{16}$$

und die Zeichenkette

$$ACATGAAC \tag{17}$$

die gleichen Häufigkeiten besitzt wie die oben definierten Wahrscheinlichkeiten und kodiert so dargestellt werden kann:

$$10110010001101. \tag{18}$$

14 Binärziffern werden gebraucht um 8 Symbole darzustellen, der Durchschnitt ist $14/8 = 1.75$ Bits pro Symbol. Dies wird mit Fano-Code bezeichnet. Der Fano-Code hat die Eigenschaft, dass wir ihn dekodieren können, ohne dass Leerzeichen zwischen den Symbolen nötig sind. Normalerweise braucht man hierfür ein „Leseraster" aber in diesem Beispiel können wir ohne eines auskommen und die Lösung herausfinden. In dieser besonderen Kodierung (Gleichung (16)) unterscheidet die erste Binärziffer die Reihe, die A enthält (welche $\{A\}$ darstellt) von der Reihe die $\{C, G, T\}$ darstellt und mit $\{A\}$ gleichwahrscheinlich ist, da $\frac{1}{2} = \frac{1}{4} + \frac{1}{8} + \frac{1}{8}$. Die zweite Binärziffer, die benutzt wird, wenn die erste Binärziffer 0 ist, unterscheidet $\{C\}$ von $\{G, T\}$. Die letzte Binärziffer unterscheidet schließlich $\{G\}$ von $\{T\}$. Da jede Auswahl gleichverteilt ist (wie in der ursprünglichen Definition von den Wahrscheinlichkeiten der Symbole), enthält jede Binärziffer 1 Bit Information. *Vorsicht!* Die letzte Aussage ist nicht immer wahr. Eine Binärziffer kann *nur* dann 1 Bit Information enthalten, wenn die beiden Reihen, die von der Binärziffer dargestellt werden, gleichwahrscheinlich sind (wie im obigen Beispiel). Sind sie nicht gleichwahrscheinlich, dann enthält eine Binärziffer weniger als ein Bit. (Erinnern wir uns, dass H maximal ist für gleichwahrscheinliche Wahrscheinlichkeiten). Wären die Wahrscheinlichkeiten die folgenden

$$P_A = \frac{1}{2}, \quad P_C = \frac{1}{6}, \quad P_G = \frac{1}{6}, \quad P_T = \frac{1}{6}, \tag{19}$$

dann gäbe es keine Möglichkeit einen (endlichen) Code zu finden, bei dem jede Binärziffer den Wert von einem Bit hätte (mit größeren Symbolreihen könnten wir dies nur annähern).[4] Beim obigen Beispiel gibt es keine Möglichkeit weniger als 1.75 Bits pro Symbol zu verwenden aber wir könnten verschwenderisch sein und weitere Binärziffern verwenden, um das Signal darzustellen. Der Fano-Code tut dies sehr gut, indem er die Symbole in einzelne Gruppen einteilt, die gleichwahrscheinlich sind; zur Fano-Kodierung finden wir weitere Hinweise in der angegebenen Literatur zur Informationstheorie. Das Maß für die Unbestimmtheit gibt einerseits an, was im idealen Fall getan werden könnte und andererseits was unmöglich ist. Das Signal mit 1.75 Bits pro Symbol kann beispielsweise nicht mit nur einer Binärziffer pro Symbol kodiert werden, das wäre unmöglich.

Bringen wir die Ideen zusammen

Zu Beginn der Einführung haben wir gesagt, dass mit dem Erhalt von Information die Unbestimmtheit sinkt. Jetzt wo wir eine allgemeine Gleichung für die Unbestimmtheit haben (8), können wir Information mit dieser darstellen und beschreiben. Nehmen wir an ein Computer habe einige Information im Speicher. Sehen wir uns nun einige Flip-Flops an, dann erhalten wir eine Unbestimmtheit mit H_{before} Bits pro Flip-Flop. Würden wir jetzt einen Teil des Speichers löschen (indem wir

[4] Dies hängt damit zusammen, dass die Brüche in der Gleichung (19) irrationale Werte annehmen wenn die Basis des Logarithmus 2 ist und da der Fano-Code durch rationale Zahlen teilt, können wir keinen Fano-Code erzeugen, der genau passt.

die Werte auf Null setzen), dann erhalten wir eine Unbestimmtheit, die kleiner ist als die vorherige: H_{after}. Der Speicher des Computers hätte dann einen durchschnittlichen Verlust von[5]

$$R = H_{before} - H_{after} \tag{20}$$

Bits von Information pro Flip-Flop. Wäre der Speicher des Computers komplett gelöscht worden, dann wäre $H_{after} = 0$ und $R = H_{before}$.

Betrachten wir nun einen Fernschreiber, der Buchstaben über eine Telefonleitung empfängt. Gäbe es kein Rauschen oder sonstige Störungen in der Telefonleitung, dann würde der Fernschreiber die Buchstaben ohne Fehler, in perfekter Weise, ausgeben. Mit Rauschen hätten wir allerdings eine Unbestimmtheit, ob der Fernschreiber das richtige Zeichen ausgibt. Bevor also ein Zeichen ausgegeben wird, muss der Fernschreiber für jedes Zeichen vorbereitet werden und dieser Zustand hat die Unbestimmtheit H_{before}, während nach jedem Zeichen das er empfängt immer noch etwas Unbestimmtheit vorhanden ist, H_{after}. Diese Unbestimmtheit basiert auf der Wahrscheinlichkeit, dass das Symbol, welches empfangen wurde nicht dem Symbol entspricht, das gesendet wurde und misst damit die Höhe (den Grad) des Rauschens.

Shannon führt dafür in Abschnitt 12 (vgl. [10] oder [13, S. 33-34]) ein Beispiel an. Ein System, das zwei gleichwahrscheinliche Symbole pro Sekunde überträgt, sendet mit einer Rate von 1 Bit pro Sekunde ohne Fehler. Nehmen wir an, die Wahrscheinlichkeit eine 0 zu empfangen, wenn eine 0 gesendet wurde, ist 0.99 und die Wahrscheinlichkeit eine 1 zu empfangen ist 0.01. „Diese Zahlen werden vertauscht, wenn eine 1 empfangen wird“ [19, S. 43]. Dann ist die Unbestimmtheit, wenn wir ein Symbol empfangen haben $H_{after} = -(0.99 \log_2 0.99 + 0.01 \log_2 0.01) = 0.081$ und die aktuelle Übertragungsrate ist $R = 1 - 0.081 = 0.919$ Bits pro Sekunde.[6]. Die Höhe der Information, die durchkommt, ist gegeben durch das Sinken der Unbestimmtheit, Gleichung (20).

Unglücklicherweise haben viele Leute den Fehler gemacht genau diesen Aspekt nicht zu verstehen und zu berücksichtigen. Daraus sind weitere Irrtümer entstanden, weil viele Leute implizit davon ausgegangen sind, dass es kein Rauschen in der Kommunikation gibt. Ist kein Rauschen vorhanden, dann ist $R = H_{before}$, wie beim Speicher eines Computers, der komplett gelöscht wurde. Das bedeutet, *wenn es kein Rauschen gibt, dann ist die Höhe der kommunizierten Information gleich der Unbestimmtheit vor der Kommunikation.* Ist aber Rauschen vorhanden, und irgend jemand geht davon aus, dass es kein Rauschen gäbe, dann führt dies dazu, dass alle Ansätze durcheinander gebracht werden. Deshalb müssen wir immer das Rauschen mit einberechnen.

Eine letzte feinsinnige Anmerkung Im letzten Abschnitt finden wir es vielleicht komisch, dass ich den Ausdruck „Flip-Flop“ verwendet habe. Dies hängt damit zusammen, dass ich unbedingt das Wort „Bit“ vermeiden wollte. Dies hat den Grund, dass das Wort zwei verschiedene Bedeutungen hat, wie weiter oben, bei der Diskussion des Fano-Codes bemerkt wurde. Die beiden Bedeutungen sollten strikt auseinander gehalten werden. Im Folgenden werden kurz die beiden Bedeutungen für „Bit“ erläutert. Ein „Bit“ kann sein:

1. Eine Binärziffer, 0 oder 1. Dies kann nur eine Ganzzahl sein. Bei diesen „Bits“ handelt es sich um die Einzelteile von Daten in Computern.

[5]Das Symbol R wurde von Shannon eingeführt. Es steht für die Übertragungsrate der Information.

[6]Shannon nutzte die Schreibweise $H_y(x)$ und meinte damit die bedingte Unbestimmtheit des Empfängers y für eine Nachricht, die x sendet, was wir mit H_{after} bezeichnet haben. Er nutzte auch den Term „Äquivokation“ (*equivocation*)

2. Ein *Maß* für Unbestimmtheit, H oder Information R. Da es sich hier um einen Durchschnitt handelt, kann es jede reelle Zahl annehmen. Es ist das Maß, welches Shannon benutzt hat, um Kommunikationssysteme zu beschreiben.

Literatur

[1] T. D. Schneider, G. D. Stormo, L. Gold, and A. Ehrenfeucht. Information content of binding sites on nucleotide sequences. *J. Mol. Biol.*, 188:415–431, 1986. http://www.ccrnp.ncifcrf.gov/~toms/paper/schneider1986/.

[2] T. D. Schneider. Information and entropy of patterns in genetic switches. In G. J. Erickson and C. R. Smith, editors, *Maximum-Entropy and Bayesian Methods in Science and Engineering*, pages 147–154, Dordrecht, The Netherlands, 1988. Kluwer Academic Publishers.

[3] T. D. Schneider and G. D. Stormo. Excess information at bacteriophage T7 genomic promoters detected by a random cloning technique. *Nucleic Acids Res.*, 17:659–674, 1989.

[4] T. D. Schneider and R. M. Stephens. Sequence logos: A new way to display consensus sequences. *Nucleic Acids Res.*, 18:6097–6100, 1990. http://www.ccrnp.ncifcrf.gov/~toms/paper/logopaper/.

[5] N. D. Herman and T. D. Schneider. High information conservation implies that at least three proteins bind independently to F plasmid *incD* repeats. *J. Bacteriol.*, 174:3558–3560, 1992.

[6] P. P. Papp, D. K. Chattoraj, and T. D. Schneider. Information analysis of sequences that bind the replication initiator RepA. *J. Mol. Biol.*, 233:219–230, 1993.

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

[8] T. D. Schneider. Sequence logos, machine/channel capacity, Maxwell's demon, and molecular computers: a review of the theory of molecular machines. *Nanotechnology*, 5:1–18, 1994. http://www.ccrnp.ncifcrf.gov/~toms/paper/nano2/.

[9] P. K. Rogan and T. D. Schneider. Using information content and base frequencies to distinguish mutations from genetic polymorphisms in splice junction recognition sites. *Human Mutation*, 6:74–76, 1995. http://www.ccrnp.ncifcrf.gov/~toms/paper/colonsplice/.

[10] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Tech. J.*, 27:379–423, 623–656, 1948. http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html.

[11] J. R. Pierce. *An Introduction to Information Theory: Symbols, Signals and Noise*. Dover Publications, Inc., New York, 1980.

[12] W. Sacco, W. Copes, C. Sloyer, and R. Stark. *Information Theory: Saving Bits*. Janson Publications, Inc., Dedham, MA, 1988.

[13] N. J. A. Sloane and A. D. Wyner. *Claude Elwood Shannon: Collected Papers*. IEEE Press, Piscataway, NJ, 1993.

[14] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., N. Y., 1991.

[15] D. K. C. MacDonald. Information Theory and Its Applications to Taxonomy. *J. Applied Phys.*, 23:529–531, 1952.

[16] M. Tribus. *Thermostatics and Thermodynamics*. D. van Nostrand Company, Inc., Princeton, N. J., 1961.

[17] S. G. Brush. Boltzmann's "Eta Theorem": Where's the Evidence? *American Journal of Physics*, 35:892, 1967.

[18] S. Hjalmars. Evidence for Boltzmann's H as a capital eta. *American Journal of Physics*, 45:214–215, 1977.

[19] Claude E. Shannon. *Ausgewählte Schriften zur Kommunikations- und Nachrichtentheorie*. Brinkmann und Bose, Berlin, 2000.

Anhang: Kleines Tutorial zu Logarithmen

Die Log-Funktion verstehen

Bei der Addition nehmen wir zwei Zahlen und addieren diese, um eine dritte Zahl zu erhalten:

$$1+1=2. \tag{21}$$

Diese Operation können wir wiederholen:

$$1+1+1=3. \tag{22}$$

Die Multiplikation ist die mathematische Operation, die dieses erweitert:

$$3\times 1=3. \tag{23}$$

Auf die gleiche Weise können wir die Multiplikation wiederholen:

$$2\times 2=4. \tag{24}$$

und

$$2\times 2\times 2=8. \tag{25}$$

Die Erweiterung der Multiplikation ist das Potenzieren:

$$2\times 2=2^2=4. \tag{26}$$

und

$$2\times 2\times 2=2^3=8. \tag{27}$$

Gelesen wird dies „zwei hoch drei ist acht“. Weil das Potenzieren einfach die Zahlen einer Multiplikation zählt, werden die Exponenten addiert:

$$2^2\times 2^3=2^{2+3}=2^5. \tag{28}$$

Die Zahl 2 ist hier die Basis des Potenzierens. Erhöhen wir den Exponent mit einem weiteren Exponent, werden diese multipliziert:

$$\left(2^2\right)^3=2^2\times 2^2\times 2^2=2^{2+2+2}=2^{2\times 3}=2^6. \tag{29}$$

Die Exponentialfunktion $y = 2^x$ zeigt der folgende Graph [7]:

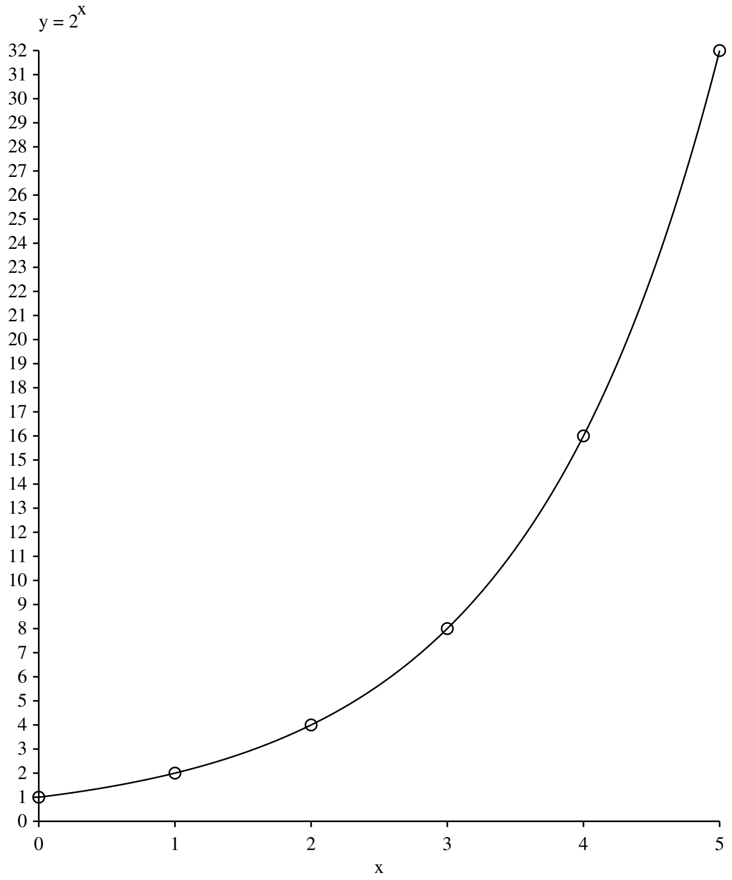

Abbildung 2: The exponential function.

Nehmen wir jetzt an, wir möchten wissen wie oft 2 mit sich selbst multipliziert werden muss, um eine bestimmte Zahl zu erhalte. Nehmen wir z. B. 2 als Basis. Wie oft muss 2 mit sich selbst multipliziert werden, um 32 zu erhalten? D. h. wir wollen die folgende Gleichung lösen:

$$2^B = 32. \tag{30}$$

Natürlich, $2^5 = 32$, also ist $B = 5$. Um dafür eine Formel zu haben, erfanden Mathematiker eine neue Funktion, den Logarithmus.

$$\log_2 32 = 5. \tag{31}$$

Gesprochen: „der Logarithmus zur Basis 2 von 32 ist 5“. Das ist die inverse Funktion zur Exponentialfunktion:

$$2^{\log_2 a} = a \tag{32}$$

and

$$\log_2(2^a) = a. \tag{33}$$

Die Logarithmusfunktion $y = \log_2 x$ zeigt der folgende Graph[8]:

[7] Das Programm, mit dem der Graph erzeugt wurde kann unter http://www.lecb.ncifcrf.gov/∼toms/delila/expgraph.html gefunden werden

[8] Das Programm, mit dem der Graph erzeugt wurde, kann unter http://www.lecb.ncifcrf.gov/∼toms/delila/expgraph.html gefunden werden

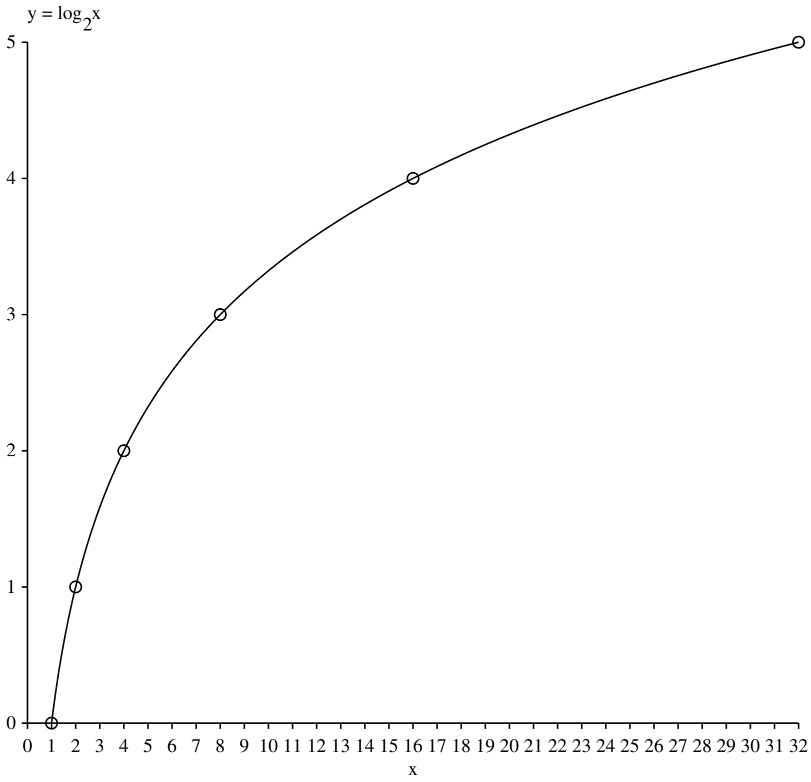

Abbildung 3: The logarithmic function.

Der Graph wurde erzeugt, indem die x- und y-Achse der Exponentialfunktion vertauscht wurden, was dasselbe ist, wie die Kurve an einer 45°-Achse zu spiegeln. Beachte insbesondere, dass $log_2 1 = 0$ und $log_2 0 = -\infty$ ist.

Das Additionsgesetz

Betrachten wir die Gleichung:

$$\boxed{2^{a+b} = 2^a \times 2^b} \tag{34}$$

was einfach die Verallgemeinerung der Gleichung (28) darstellt. Nehmen wir an beiden Seiten der Gleichung den Logarithmus:

$$\log_2 2^{a+b} = \log_2 \left(2^a \times 2^b\right) \tag{35}$$

Potenzieren und der Logarithmieren sind also inverse Operationen, wir können die linke Seite kürzen:

$$a + b = \log_2 \left(2^a \times 2^b\right) \tag{36}$$

Jetzt wird es trickreich, wir ersetzen $\log_2 x = a$ und $\log_2 y = b$:

$$\log_2 x + \log_2 y = \log_2 \left(2^{\log_2 x} \times 2^{\log_2 y}\right) \tag{37}$$

Nochmal, Potenzieren und Logarithmieren sind inverse Opertionen, wir könne die beiden Fälle auf der rechten Seite kürzen:

$$\boxed{\log_2 x + \log_2 y = \log_2 (x \times y)} \tag{38}$$

An dieser additiven Eigenschaft war Shannon interessiert.

Die „Weiterleitungsregel“ (pull-forward rule)

In der Gleichung (32):

$$a = 2^{\log_2 a}. \tag{39}$$

Erhöhen wir beide Seiten mit u:

$$a^u = \left(2^{\log_2 a}\right)^u. \tag{40}$$

Dann können wir die Exponenten verbinden, in dem wir wie in (29) multiplizieren:

$$a^u = 2^{u \log_2 a}. \tag{41}$$

Schließlich nehmen wir den Logarithmus zur Basis 2 auf beiden Seiten und kürzen die rechte Seite:

$$\boxed{\log_2 a^u = u \log_2 a} \tag{42}$$

Diese Regel können wir uns merken, als eine, die es uns erlaubt, den Exponenten aus dem inneren des Logarithmus nach außen zu „leiten“.

Wie wir die Basis eines Logarithmus konvertieren können

Taschenrechner und Computer können in der Regel den Logarithmus zur Basis 2 nicht direkt berechnen. Wir können aber einen Trick nutzen, der uns die Berechnung vereinfacht. Beginnen wir mit der Gleichung:

$$x = \log_z a / \log_z b \tag{43}$$

Arrangieren dies neu:

$$\log_z a = x \log_z b. \tag{44}$$

Und nutzen jetzt die „umgekehrte Weiterleitungsregel“(!):

$$\log_z a = \log_z b^x \tag{45}$$

entfernen die Logarithmen:

$$a = b^x. \tag{46}$$

nehmen dann den Logarithmus zur Basis b:

$$\log_b a = \log_b b^x. \tag{47}$$

Vereinfachen dieses zu:

$$\log_b a = x. \tag{48}$$

Da wir aus der Gleichung (43) den Wert von x kennen:

$$\log_b a = \log_z a / \log_z b \tag{49}$$

Die Konversionsregel, um den Logarithmus zur Basis 2 von jeder Basis z zu erhalten, lautet:

$$\boxed{\log_2(a) = \log_z(a)/\log_z(2)} \tag{50}$$

Beachte: da z nicht auf der linken Seite erscheint, ist es völlig egal, welchen Logarithmus wir zur Verfügung haben, weil wir mit der folgenden Gleichung immer den Logarithmus einer anderen Basis berechnen können: Versuchen wir das folgende Beispiel auf einem Taschenrechner aus:

$$\log_2(32) = \frac{\log_{\text{whatever!}}(32)}{\log_{\text{whatever!}}(2)}. \tag{51}$$

Wir sollten 5 erhalten.

Tricks With Powers of 2

In der Analysis lernen wir den natürlichen Logarithmus mit der Basis $e = 2.718281828459045\ldots$[9] Berechnungen können mit den Taschenrechner oder mit dem Computer sehr einfach durchgeführt werden, aber die meisten haben Schwierigkeiten, mit dieser Zahl im Kopf zu rechnen. Dagegen können wir uns an die Potenzen zur Basis 2 leicht erinnern:

choices M	bits B
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1024	10

mit $2^B = M$ und $\log_2 M = B$.

Diese Tabelle können wir nutzen, um schnell Abschätzen der Logarithmen von größeren Zahlen vorzunehmen. Beachte dass:

$$2^{10} = 1024 \approx 1000 = 10^3. \tag{52}$$

[9] Möchten Sie Ihre Freunde beeindrucken, indem Ihr Euch an diese Zahl erinnert? Dann merkt Euch: nach der 2.7 (das müsst Ihr Euch selbst merken!) folgt zweimal 1828 und dann ein 45°-90°-45°Dreieck.

Wir nehmen den Logarithmus zur Basis 2 von 4×10^6, und denken uns:

$$\begin{aligned}
\log_2(4 \times 10^6) &= \log_2(4) + \log_2(10^6) && (53)\\
&= 2 + \log_2(10^3 \times 10^3) && (54)\\
&= 2 + \log_2(10^3) + \log_2(10^3) && (55)\\
&\approx 2 + \log_2(2^{10}) + \log_2(2^{10}) && (56)\\
&\approx 2 + 10 + 10 && (57)\\
&\approx 22 && (58)
\end{aligned}$$

Der korrekte Wert lautet 21.93