

# The Cost of AI

Matt Mahoney  
Draft, Mar. 27, 2013

## Abstract

In 2011, we paid people worldwide US \$70 trillion to do work that machines did not know how to do. Automating the global economy will require solving hard problems in language, vision, robotics, art, and modeling human behavior. We estimate the computational costs to be  $10^{26}$  operations per second,  $10^{25}$  bits of memory,  $10^{19}$  input/output bits per second, and  $10^{17}$  bits of human knowledge collected at a rate of 7 bits per person per second. Lowering the total cost below the break-even point of \$1 quadrillion will require a  $10^5$  fold improvement in both the manufacturing cost and energy efficiency of computation, which is unlikely to be achieved by further shrinking transistor sizes, and by a global cultural acceptance of the loss of privacy over a period of decades. Software development is not a significant contributor to the cost of AI because a human baby has a Kolmogorov complexity equivalent to only  $10^8$  to  $10^9$  lines of code.

## Introduction

We estimate the cost of automating human labor worldwide. We assume that any technical solution will require computing power approximately equivalent to the world population of 7 billion human brains, and its complexity will be of the order of the sum of human knowledge. Each of these far exceeds what is currently available, which we offer as an explanation for the failure (so far) of artificial intelligence (AI).

The complexity of humanity has two parts. Humans store about  $10^9$  bits of information in their DNA and another  $10^9$  bits in high-level long term memory, but the latter varies more from person to person, and collectively makes up most of the knowledge that machines need to know to do what we do. This knowledge far exceeds what is available on the internet, and must be extracted through slow channels like speech and writing. Assuming the cost of hardware drops, the time spent by humans providing this information will dominate the cost of AI.

One could argue that intelligence does not require human knowledge. It depends on what you mean by “intelligence”. Although we use the term “AI”, we make explicit that the goal is to create machines that do what you want, not just what you tell them. Successful

communication between agents requires that each be able to guess what the other knows or doesn't know. This requires that machines have models of the minds of the people they communicate with. A model is a function that takes sensory input and returns a prediction of your actions. There is a strong economic incentive to develop models of yourself and others. A model could be used in simulations to predict what would make you happy, or what would make you buy something.

An immediate consequence of AI, and therefore a secondary goal, is life extension by repairing or replacing failed body parts, including the brain. We would probably have no objections to restoring function lost to stroke, injury, or Alzheimer's disease by replacing brain tissue or neurons with functionally equivalent devices. Likewise, your entire brain could be replaced with a computer programmed to carry out the predictions of your model in real time and placed back in your body or that of a robot, and nobody would notice any difference. Such an "upload" would be effectively immortal because your memories could be backed up periodically and copied to another robot in case of an accident.

Humans, like all animals, have brains programmed by evolution to fear the things that can kill them, residing in bodies programmed to grow old and die. Therefore, uploading must be done in a way that does not arouse this fear. You see your friends go in for a procedure and come out younger, stronger, healthier, smarter, and happier. You might not accept this procedure if it involves presenting you with a robot that looks and acts like you, and then asking you to shoot yourself. It would be more acceptable if microscopic robots gradually replaced your cells with equivalent devices without you noticing any change, even if the end result is exactly the same. The essential requirement seems to be that there is not the appearance of two copies of you active at the same time. Hayworth's (2010) proposal of destructively scanning the brain prior to programming a robotic copy might be acceptable if the alternative is dying without collecting this data.

## Requirements for AI

In order for machines to do the work of humans, they must be able to do any of the following as well as humans:

- Converse and answer questions given in natural language speech or writing.
- Predict missing letters or words in text.
- Given a bilingual dictionary and 1 GB of monolingual text in a new language, learn to translate from one to the other.
- Translate speech to text.
- Translate text to speech with proper inflection.
- Design, write, test, and debug software given a natural language specification.
- Pass college level final exams in any subject.
- Predict the recommendations of referees for journal paper submissions in any field of

research.

- Recognize when two texts are by the same author based on content and style.
- Recognize common sounds.
- Translate images of written words to text.
- Recognize common objects in pictures or video.
- Recognize if two images shown in succession are of the same person.
- Recognize if two speech signals are spoken by the same person.
- Match videos to scripts or written descriptions.
- Recognize human emotions from facial expressions, tone of voice, and context.
- Predict the effects of text, images, and video on human emotions (funny, sad, outrage, excitement, sexual arousal, etc.), and therefore be able to produce art, humor, entertainment, and pornography by iterative search.
- Identify music by genre and artist and rate its quality (thus reducing music generation to an iterative search process).
- If equipped with an arm, pick up, throw, catch, or place an object on command.
- If equipped with legs or wheels, navigate to a given location on command over roads or rough terrain.
- Learn to predict people's actions while watching or interacting with them.

There is no requirement that an AI be autonomous. There is no requirement that an AI have (as opposed to recognize) emotions, feelings, or goals. There is no requirement that it be “conscious” or “sentient”, and therefore no need to define these terms. We explicitly define intelligence (the “I” in “AI”) as the ability to pass the tests listed above.

Uploading requires realistic looking humanoid robotic bodies and the ability to model specific humans with enough fidelity to fool others. It differs from automating work in that it requires a single machine with all of these capabilities, rather than a large number of specialized machines such that for each capability, there is at least one machine that satisfies it. Nevertheless, the list of requirements is essentially the same.

## Hardware Costs

We assume that a human brain sized neural network is required. We do not know this with certainty, but we do know that the best known solutions to hard problem like vision and language use algorithms based on neural networks that run on thousands of processors, for example (Ferrucci, 2010; Gorrell, 2006; Quoc, 2012) based on principles described in (Rumelhart and McClelland, 1986). We also know that large brains have a high energy cost, and that evolution so far has failed to find a way to produce human level intelligence with insect sized brains after billions of years. It would be arrogant for us to believe that we are smarter than evolution while we are still susceptible to aging, death, and disease.

The human brain has about  $10^{11}$  [neurons](#) and  $10^{14}$  to  $10^{15}$  synapses. More precisely, the cerebral cortex makes up 19% or  $1.6 \times 10^{10}$  neurons out of a total of  $8.6 \times 10^{10}$  (Azevedo et.

al., 2009). These have an average of 7000 synapses each (Drachman, 2005), for a total of  $1.1 \times 10^{14}$  synapses. Most of the neurons are located in the [cerebellum](#), which makes up only 10% of the brain volume and is responsible for fine motor skills. This is due mainly to the  $5 \times 10^{10}$  small granule cells with 80-100 connections each to Purkinje cells for a total of  $4-5 \times 10^{12}$  connections. In addition, another  $2 \times 10^9$  mossy fibers form 500 connections each to granule cells for a total of  $10^{11}$  connections. Thus, the vast majority (96%) of synapses are found in the cerebral cortex, which is associated with higher level thought, perception, and action.

In the most widely accepted neural models, information is carried by the spiking rate, which can range from 0 to 300 per second, rather than the spikes themselves. We may assume an information rate on the order of 10 to 100 bits per second. The basic operations are computing the firing rate as a function of the weighted sum of inputs, and updating the synaptic weights as a function of the input and output neuron firing rates over time. Thus a simulation requires on the order of  $10^{15}$  bits (1 petabit) using a few bits to represent a synapse, and  $10^{16}$  operations per second (10 petaflops). To do the work of all  $10^{10}$  humans would require  $10^{25}$  bits and  $10^{26}$  operations per second.

A [human retina](#) has 75 to 150 million rods and cones that transmit on the order of 10 bits per second. Duplicating just the vision of  $10^{10}$  people represents about  $10^{19}$  input pixels per second.

[Moore's Law](#) is an observation that the cost of computing power drops by  $\frac{1}{2}$  about every 1.5 or 2 years. At the current rate, the cost of both CPU and memory would drop below US \$1 quadrillion in the 2030's. This would be competitive with the global value of human labor (GDP divided by market interest rates). Note that if the hardware requirement is off by a factor of 10, then it does not change the cost, but instead changes the time to AI by 5 to 7 years.

A typical supercomputer uses  [\$10^{-9}\$  Joule](#) per operation, as do smaller computers. By contrast, human energy consumption is about 2500 Kcal per day, or 100 Watts, of which 25 W is used by the brain. This is  $10^5$  times as energy efficient as silicon. This efficiency is unlikely to be achieved by further shrinking chip feature sizes, which are currently around 22 nm or about 100 silicon atoms. At the current cost of electricity of about \$0.10/kWh, human brain equivalent computation would require 10 MW and cost \$1000 per hour, which is not competitive with human labor. Running  $10^{10}$  such computers, assuming we could, would produce  $10^{17}$  W of waste heat, equal to 60% of the energy received from space as sunlight. Dissipating this much energy would raise the Earth's average temperature by a factor of  $1.6^{0.25} = 1.125$ , or from 15 C to 51 C (from 59 F to 123 F).

## Software Costs

We wish to estimate the software complexity (lines of code and cost) of AI. We will estimate that a line of code costs \$100 to write at a rate of 10 to 20 lines per day per developer.

AI requires both a brain and a body. Therefore, we should expect its [algorithmic \(Kolmogorov\) complexity](#) to be similar to that of a human. The instructions for creating a human baby are encoded in our [DNA](#), which has a haploid count of  $3 \times 10^9$  base pairs or  $6 \times 10^9$  bits. This is an upper bound on information content. Compressing the genome can reduce this bound slightly. Using the best known data compressors on the human reference genome and making some reasonable assumptions given additional computing resources, we can estimate that the information content of the human genome is no more than  $4.58 \times 10^9$  bits (Appendix A).

To estimate the complexity of a line of code, we again use the best known compression methods to compress 927K lines (30 MB) of C source code from [gimp](#) v2.0.0 (2004), a graphics editor, and header files from [mingw](#) 4.5.0 (2010), a C++ compiler. The result is an upper bound of 16 bits per line of code (Appendix A). Equating the two, we estimate that the human genome is similar in complexity to 300 million lines of code, or \$30 billion.

We should note that the true complexity of the human genome is not known. There is no general algorithm for computing algorithmic complexity. However, the table suggest that DNA is harder to compress than source code. Therefore, the use of better compressors to improve accuracy is likely to raise the estimated cost.

One may argue that the genome has a much lower complexity because the [exome](#), the part that encodes genes, makes up only 1.5% of the total. We do not fully understand the role of the remaining DNA, or how much of it is important. We may therefore approximate a lower bound by studying the [genome size variation](#) of other species. There is a wide variation even among related species, but we observe that the minimum size tends to increase consistently from lower to higher organisms. We assume that there is genetic pressure in some species toward smaller genomes (which can reproduce faster), and therefore that drastically smaller sizes are not possible. The smallest genome for mammals is about  $2 \times 10^9$  base pairs.

## Knowledge Collection Costs

We have so far estimated the cost of building and programming a baby AI. It is often argued that you only need to train an AI once to bring it up to college level, and then you can make billions of copies of the knowledge for free. That may be true, but what we wish to estimate is the cost of giving each AI the specific knowledge that is unique to its job from that point forward.

We do not expect AI robots to replace humans 1 to 1. Rather, it will be more usual for one machine to do part of the work of many people. This will not change our estimate because we are only interested in the total amount of knowledge needed to do everything that people now do, regardless of how the work is redistributed.

AI requires human knowledge, that is, things that people know. Human communication is successful when both parties can correctly guess what the other person knows and doesn't

know. Human-machine interfaces often fail because the computer does not have an accurate model of your mind. It cannot predict your responses to its outputs.

Landauer (1986) estimated that human long term memory capacity is  $10^9$  bits, as measured by recall tests for words, pictures, and music clips. This would be  $10^{19}$  bits for  $10^{10}$  people, except that most of this knowledge is shared or written down somewhere, and therefore easily copied to an AI. But let us assume that 1% to 10% of what you know is not written down or known to anyone else, leaving  $10^{17}$  to  $10^{18}$  bits that makes each human mind unique. We assume that most of what you know is either relevant to your work or it influences your purchasing or business decisions, possibly indirectly. Thus, this is the approximate algorithmic complexity of the global economy.

We cannot collect this information from the internet. A quick Google search for common words like “a” and “the” reveals about  $2.5 \times 10^{10}$  web pages in 2012. If we assume  $10^4$  bits per page after removing duplicates and compression, then only 0.1% of human knowledge is readily available. To illustrate the impact, if a robot were to start cleaning your house, it would not know which items should be saved or thrown away until you tell it, unless you wrote down that information in advance. The cost of AI is the time you spend training the otherwise intelligent robot, multiplied by 7 billion people.

The U.S. Labor Dept. estimates that it costs \$15,000 to replace an employee, or 1% of lifetime earnings. The [cost varies widely with skill level](#), ranging from \$3500 for a job paying \$8 per hour, to 1.5 years salary for middle level managers, to 4 years salary for top level employees. A major factor is the cost of re-learning what the old employee knew, but did not write down, like what you know about the people you work with. This knowledge is unique to each person, even for people with the same job description at the same company. The average cost will rise as the low skilled jobs are automated first.

Human knowledge must either be collected through slow channels like speech and typing, or by high resolution brain scanning using technology yet to be developed. Shannon (1950) estimated that written English has an information content of about 1 bit per character, which is in agreement with the [best text compressors](#). Spoken English, such as the [Switchboard Corpus](#), is about half this rate, based on studies of [language models for speech recognition](#). At 150 words per minute, 5.5 characters per word including spaces, speech has an information rate of 7 bits per second or 25K bits per hour. Typing at 75 words per minute has the same rate. The global average wage rate is \$5 per hour assuming 2000 hours per year. Thus, the cost of collecting  $10^{17}$  to  $10^{18}$  bits is \$20 trillion to \$200 trillion.

The cost of knowledge collection could be reduced by using surveillance to learn about you by observation while you do other things. This would include recording everything you do on a computer, something we have already started doing. Alternatively, this information could be collected by high resolution brain scanning using technology yet to be developed, provided the cost were less than \$3000 to \$30,000 per person. I don't believe this is likely to happen

before 2030.

The total cost of AI will be dominated at first by hardware, and then later by the cost of human knowledge. The software cost, although substantial, will be an insignificant fraction. We will spend additional software effort at first to optimize for slow hardware, and then later to compensate for incomplete human knowledge.

## Alternative Complexity Measures

The absolute measure of information, up to a language-dependent constant, is Kolmogorov complexity, or the length of the shortest program which outputs this data. In general, this value is not computable, but can only be bounded from above by the shortest *known* program. Furthermore, for the purpose of estimating cost, we wish to use the shortest known program that can be computed with feasible resources. In Table 2, we consider 4 possible estimates of the complexity of human civilization based on different algorithms for producing it, and estimate the cost (in bit operations and bits of memory) to run the algorithm. Then we explain how these numbers were derived.

Table 2. Cost estimates of four approaches to AI.

Algorithm	Complexity (bits)	Operations	Memory (bits)
Engineered	$10^{17}$	$10^{36}$	$10^{25}$
Evolution	$10^7$	$10^{49}$	$10^{37}$
Cosmology	$10^3$	$10^{120}$	$10^{120}$
Multiverse	$10^0$	$10^{240}$	

The engineering approach is the one just described, run for the average age of a human, 30 years =  $10^9$  seconds. It consists of building fast and energy efficient computers using technology yet to be developed, and collecting, publishing, and making searchable everything you say and do in order to develop a public model of your mind. In this model, the internet will become a “global brain” to which you can post messages to a permanent global pool, and they are sent to anyone who cares, human or machine. I described one possible design in my [proposal for distributed AI](#). I believe that public surveillance will be acceptable because it is two-way. Queries and responses are both public, just like with face to face communication. I cannot learn anything about you without you knowing that I am asking.

### Evolutionary model

Evolution is a learning algorithm that adds information to the genome at a maximum rate of

log n bits per generation of n children per parent. We may estimate the information content of the human genome by comparing it to the chimpanzee, which diverged from humans 6 million years ago and shares [96% of our DNA](#), or all but  $1.2 \times 10^8$  base pairs. Chimpanzees reproduce from about age [9 to 40](#). If we assume a total of  $10^6$  generations for both species, then we would conclude that the effective information content of DNA is at most 0.008 bits per base pair, or less due to parallel evolution. Thus, the human genome would contain at most  $3 \times 10^7$  bits of information.

Evolution is a search algorithm for strings x that maximize the unknown function fitness(x). The search proceeds by copying x in parallel and making minor random edits by inserting, deleting, or modifying DNA bases or fragments, or in the case of sexual reproduction, taking fragments from two other strings. We can think of DNA copying, RNA transcription, and protein synthesis as elementary operations per base.

The world [biomass](#) consists of about  $10^{31}$  cells (mostly bacteria and plants, and  $10^{22}$  human cells) with an average of  $10^6$  DNA bases per cell, or  $10^{37}$  bases. Each base represents 2 bits of memory. Global carbon production is  $1.2 \times 10^{17}$  g =  $5 \times 10^{39}$  atoms per year =  $1.5 \times 10^{32}$  atoms per second (Vernadsky, 1998, p. 72). The evolution of humans took  $10^{17}$  seconds (3 billion years) from the origin of life, for a total of  $10^{49}$  operations.

Freitas (2000) examined the capacity of self-replicating nanotechnology as artificial life. Robots cannot be much smaller or reproduce much faster than bacteria due to the energy needed to move atoms. However, there is room for improvement. Global carbon production by photosynthesis uses  $1.33 \times 10^{14}$  W (Vernadsky) or 0.15% of the [8.9 x 10<sup>16</sup> W](#) of solar power that reaches the Earth' surface. Also, each operation uses  $1.1 \times 10^{-18}$  J, which is 400 times the thermodynamic limit (Lloyd, 2000) of  $kT \ln 2 = 2.8 \times 10^{-21}$  J per bit operation at 290 K.

Simulating human evolution in silicon is not feasible. The [world's most powerful supercomputers](#) in 2012 execute  $10^{16}$  operations per second using  $10^7$  W, or  $10^{-9}$  J per operation. This is  $10^9$  times higher than biology. [Global energy consumption in 2010](#) from oil, coal, gas, nuclear, and other sources was  $1.8 \times 10^{13}$  W, or 1/7 of the power used by plants. Furthermore, simulating chemistry requires solving the Schrodinger equation, which has exponential time complexity in the number of particles unless it is run on a quantum computer.

## Cosmological model

An alternative way to describe human civilization would be to describe the laws of physics (a few hundred bits) and the initial state of the universe at the Big Bang (presumably simple), and simulate the observable universe. Optionally, one could add 80 bits to describe which of  $10^{24}$  planets we evolved on, in case life evolved elsewhere. Lloyd (2001) estimated that such a computation would require  $10^{120}$  operations and  $10^{90}$  bits of memory on a quantum computer, or  $10^{120}$  bits if quantum gravity effects are included. This is also the computational



capacity of the universe, and therefore such a computation would require an even larger computer. This is consistent with Wolpert's theorem (2001), which states that two computers cannot mutually simulate or predict each other's output. Since this also applies if the computers are identical, it means that a computer cannot simulate itself.

Quantum computation is time-reversible, and therefore not subject to thermodynamic costs, unlike irreversible operations like copying DNA or transcription. However, there is a recoverable energy cost of  $E = h/4t$ , where  $t$  is the time to perform a qubit flip and  $h$  is Planck's constant =  $6.626 \times 10^{-34}$  Joule-seconds. Converting all of the observable universe's mass of  $3 \times 10^{54}$  kg into energy by  $E = mc^2$  allows  $10^{120}$  operations since the time of the Big Bang 13.7 billion years ago.

The memory capacity of  $10^{90}$  bits is estimated by encoding information by the position and velocity of the approximately  $10^{80}$  particles in the universe within the limits of the Heisenberg uncertainty principle. The larger figure of  $10^{120}$  is given by the [Bekenstein bound](#) of  $A/(4 \ln 2)$  bits, where  $A$  is the area in Planck units,  $hG/2\pi c^3 = 2.612 \times 10^{-70}$  m<sup>2</sup>. The exact value depends on the mass and size of the universe. For a black hole with a radius of [13.8 billion light years](#), the entropy would be  $2.95 \times 10^{122}$  bits, making each bit about the size of a proton.

### Multiverse model

The multiverse model is the simplest, and therefore the most likely by the principle of Occam's Razor. It supposes that all possible universes were enumerated, and that the laws of physics that we observe are the result of our existence being possible. For example, if the ratio of the masses of the proton and neutron were slightly different, then hydrogen fusion in stars would not occur, or supernova explosions would have produced the wrong ratio of elements for life to evolve.

We might suppose a [Levin search](#), where the  $n$ 'th possible universe is run for  $n$  steps. Since our universe requires  $10^{120}$  steps, it would be about the  $10^{120}$ th possible universe and therefore it would take  $10^{240}$  steps to reach this point. Furthermore, it means that our universe has a description length of  $\log_2 10^{120} = 400$  bits.

I did not estimate memory requirements. If we assume that alternate universes are simulated in parallel, then the memory requirement would be  $10^{240}$  bits. However, that assumes the existence of time, which is a property of some (but not all) possible laws of physics. A multiverse is a purely mathematical object.

## Implications of Expensive AI

**AI development and ownership will be globally distributed over the internet.** AI will be too expensive for any person or company to own or control. AI will consist of lots of narrow experts who can either answer questions in their area of expertise, or know who to ask. The

human owner of each agent will have a vested interest in disseminating its knowledge and protecting its reputation in competition with other experts.

**AI will look like a global brain.** Agents will communicate so fast that to us they will all appear to have the same knowledge. When you ask a question or post a message, it will be routed to anyone who cares, whether it be human or machine. In my [thesis](#) and distributed AI proposal, messages go into a globally readable and indexed pool and cannot be deleted. I show that  $n$  bits of distributed knowledge can be indexed in roughly  $O(n \log n)$  space with searches and updates in  $O(\log n)$  time by a distributed index. Routing is achieved by agents trading messages in an economic model in which information has negative value. Agents mutually benefit by accepting messages which they can compress better, i.e. are semantically similar to what they already know, and remembering who sent them.

**Privacy will end.** The least expensive way to collect human knowledge is by observation. Moore's Law will make it inexpensive to have your phone and high resolution webcams and microphones everywhere broadcasting onto the internet, where other agents can recognize faces and speech and make it instantly searchable. People will willingly broadcast every detail of their lives, and pay to do so, as long as surveillance is public and bidirectional. When someone searches for you by name, you will be notified. The end of secrecy will help solve the identity theft problem because nobody can pretend to be you without everyone knowing what they are doing. Publishing the data that allows others to build models of your mind is mutually beneficial. Models could predict what would make you happy, or what would make you buy something.

**AI will not cause massive unemployment.** Technology has always resulted in economic growth, a higher standard of living, longer life expectancy, and more choices in the job market. It is easy to see the jobs made obsolete by automation, but harder to see where the new jobs come from. Technology makes stuff cheaper, which leaves money left over to buy other stuff. That extra spending creates new jobs. Furthermore, because AI is expensive, this will happen slowly enough to adapt as the least skilled jobs are replaced first.

One problem is that in a free market, a person cannot start from nothing because AI has made any possible job skills obsolete. It is already true that in a free market, the rich get richer and the poor starve, because the rich own most of the technology needed to make money. Thus, it remains necessary to have governments that tax the rich and give to the poor. Economic growth from AI will allow a smaller tax to provide basic necessities for everyone.

**AI will not end scarcity.** AI will reduce the cost of manufacturing and computing, but not of raw materials, energy, land, and space for waste disposal. Those costs will rise in response to population growth and ultimately limit population. Immortality and reproduction are not both possible. Since 1800, there has been no Malthusian limit on population because the exponential growth of technology (with respect to the cost of food) has been faster than the exponential growth of population.

**Unfriendly AI is not a short term threat.** [Vinge](#) and [Kurzweil](#) argue that if humans can create smarter than human intelligence, then so can they, only faster. This accelerating improvement would converge quickly to unimaginable power at a point in time known as the Singularity. [MIRI](#) (formerly the Singularity Institute) was founded to address the risk of an “unfriendly” self improving AI, acting according to its own goals beyond our control.

It should be clear that it is all of humanity that creates AI, and therefore that is the threshold to be crossed. We must also define “intelligence”. Two commonly accepted tests are:

- The Turing test (Turing, 1950). A machine is intelligent if it cannot be distinguished from a human by written communication with it.
- Universal intelligence (Legg and Hutter, 2006) is the expected reward of a goal seeking agent interacting and receiving a reinforcement signals from an environment chosen at random from a universal or [Solomonoff distribution](#), i.e. favoring simpler descriptions.

By the Turing test, superhuman intelligence is impossible because nothing can be more like a human than a human, even though computers have already surpassed humans by [some tests](#). Thus, the fear is that a goal seeking AI will either have its initial goals specified incorrectly (because they are too complex to specify), or that the goals will drift as the agent modifies itself. For example, an AI told to [maximize paperclip](#) production might misinterpret its goal as it became more powerful and tile the solar system with molecule size paperclips, killing all life in the process.

Unfriendly AI is not a risk, at least in the short term, for three reasons. First, by its construction, it is a tool to increase human productivity, and not a goal seeking reinforcement learner. Second, its behavior is controlled by billions of users, so any set of behaviors it is given is more likely to be correct (or at least a consensus) of humanity than if a single person or a committee specified them. Third, it is [fundamentally impossible](#) for a program to increase its own knowledge or computing power, the two components of intelligence, by rewriting its own software. Any improvement must come from learning from its environment and building more computing hardware. Any threat depends on how fast these two things can happen.

**Self-replicating agents will be an existential threat.** Self replicators could include natural or genetically engineered organisms, intelligent computer viruses, and self replicating robots or nanotechnology. Replicators could compete with us for resources or feed on us. Replicators may evolve to improve reproductive fitness. Already we have seen computer viruses evolve (with human intervention) to feed on their hosts without killing them, just like the evolution of natural parasites. Intelligent viruses that model human behavior could trick us into installing them, or analyze and debug code to find security weaknesses.

The greatest threat is probably the accidental release of self-replicating, autonomous nanotechnology. Smaller robots can reproduce faster. Freitas (2000) concludes that the smallest feasible robot would be about the size of a bacteria or virus, and would be limited by

available energy and heat dissipation to reproduce within an order of magnitude of the same rate as biological organisms. Uncontrolled nanotechnology could displace all DNA based life in a few weeks.

Uploads are autonomous robots with human rights. Some of these may choose to self modify so that they replicate rapidly and pass on this characteristic to their offspring. Thus, uploading is a deliberately created risk.

**Maximizing happiness = death.** In the goal seeking or reinforcement model of AI, this means maximizing a utility function which depends on mental state. Normally, we can only do this by manipulating the environment. But an uploaded mind could also do this by modifying its own software. A state of maximum utility would be static. Any thought or sensory input would be unpleasant because it would result in a different state with lower utility.

It should be noted that in spite of our technology, there is no evidence that humans are happier today than 1000 years ago, or even more than other species. Suicide is rare among animals other than humans; the exceptions being whales and dolphins, both of which have larger brains than us.

**We have far to go.** Table 2 shows us that we can go far, far beyond human level AI. In the evolutionary model, the current biomass or equivalent nanotechnology will support  $10^{12}$  times as many human mind equivalents as currently exist. This number is limited by available energy from the sun. By building a [Dyson sphere](#), we could capture all  $3.846 \times 10^{26}$  W of output, enough to increase our population by another factor of  $10^{10}$ . By going to [other stars](#), we could increase our population by another factor of  $10^{23}$  for a total of  $10^{55}$  human mind equivalents and still be ahead of the physical limits of computation by a factor of  $10^{49}$ .

## References

Azevedo et. al. (2009), "[Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain](#)", *J. Comparative Neurology* 513:532-541.

Bonfield, J. K, and M. V. Mahoney (2013), "Compression of FASTQ and SAM Format Sequencing Data", *PLoS ONE* (to appear).

Drachman, D. (2005), "[Do we have brain to spare?](#)", *Neurology* 64(12).

Freitas, R. (2000), "[Some Limits to Global Ecophagy by Biovorous Nanoreplicators, with Public Policy Recommendations](#)", Foresight Institute.

Ferrucci et. al. (2010) "[The AI behind Watson](#)", *AI Magazine*.

Gorrell, G. (2006), "[Generalized hebbian algorithm for incremental latent semantic analysis](#)", *Proc. Interspeech*.

Hayworth, K (2010), "[Killed by Bad Philosophy](#)", [www.brainpreservation.org](http://www.brainpreservation.org).

hg19 (2009), [UCSC Genome Browser](#).

Hutter, Marcus (2003), "[A Gentle Introduction to The Universal Algorithmic Agent {AIXI}](#)", in *Artificial General Intelligence*, B. Goertzel and C. Pennachin eds., Springer.

IDC (2012), "[2.8 ZB of Data Created and Replicated in 2012](#)", *Storage Newsletter*.

Landauer, Tom (1986), "How much do people remember? Some estimates of the quantity of learned information in long term memory", *Cognitive Science* 10:477-493.

Legg, S. (2006), "Is there an Elegant Theory of Prediction?", [arXiv:cs/0606070v1](http://arXiv:cs/0606070v1) [cs.AI].

Legg, S., and M. Hutter (2006), "[A Formal Measure of Machine Intelligence](#)", *Proc. Annual machine learning conference of Belgium and The Netherlands (Benelearn-2006)*. Ghent.

Lloyd, Seth (2000), "Ultimate physical limits to computation", [arXiv:quant-ph/9908043v3](http://arXiv:quant-ph/9908043v3).

Lloyd, Seth (2001), "Computational Capacity of the Universe", [arXiv:quant-ph/0110141v1](http://arXiv:quant-ph/0110141v1).

Quoc et. al. (2012), "Building high-level features using large scale unsupervised learning", [arXiv:1112.6209v3](http://arXiv:1112.6209v3) [cs.LG].

Rumelhart, D. E., J. L. McClelland, and the PDP Research Group (1986), *Parallel Distributed Processing*, Cambridge MA: MIT Press.

Shannon (1950), Claude E., "Prediction and Entropy of Printed English", *Bell Sys. Tech. J* 3:50-64.

Turing, A. M., (1950) "Computing Machinery and Intelligence", *Mind*, 59:433-460.

Vernadsky, Vladimir I. (1998), *The Biosphere*, Springer.

Wolpert, D. (2001), "[Computational Capabilities of Physical Systems](#)", *Physical Review E*, 65:016128.

# Appendix A. Source Code and Human Genome Compression Results

## Compressors

To estimate information content of source code and the human genome, we used several compression programs including those among the top ranked by compression ratio on the [Silesia corpus](#), [Large Text Benchmark](#), [Maximum Compression benchmark](#), [Squeeze Chart](#), and [Compression Ratings](#) without regard to speed or memory usage. For each compressor, options are selected for maximum compression at the expense of speed and memory.

[Zip](#) 3.0 compresses in the widely used [deflate](#) format using the [LZ77](#) algorithm. Duplicate occurrences of strings are replaced with pointers to the previous occurrences. Matches and literals are [Huffman](#) coded, i.e. using variable bit length codes packed together. -9 selects maximum compression by searching longer for matches.

[7-zip](#) v9.30a uses a variant of LZ77 called [LZMA](#). It compresses better by using a larger match window and by [arithmetic coding](#) the literal and match symbols. -mx selects maximum compression.

[BBB](#) uses a memory-efficient [Burrows-Wheeler transform](#) (BWT) followed by a fast-adapting order-0 context model and arithmetic coding. A BWT sorts the input by context, which tends to produce long runs of identical or related bytes, which compress easily. BBB has a “slow” mode that requires 1.25 times the input size in memory, which is  $\frac{1}{4}$  of the normal requirement. The option “cfm30” selects fast mode (using 5x memory) and a block size of 30 MB. In all experiments, the block size is set larger than the input size.

[ppmonstr](#) variant J is the top ranked [PPM](#) compressor. It predicts characters one at a time based on the previous 32 bytes (with -o32 option), dropping to a lower order context when no previous match is found. -m1600 selects 1.6 GB of memory to store statistics. When memory is used up, some of the statistics are discarded to make room. Using a lower order conserves memory and improves compression in this case. Otherwise the highest order possible should be used.

[Nanozip](#) 0.09a with option -cc and the various PAQ compressors such as [paq8pxd](#) and [paq8px](#) v69 use [context mixing](#) algorithms. Bits are predicted one at a time and arithmetic coded. In the PAQ variants, there are hundreds of models whose predictions are adaptively averaged together, making the programs extremely slow (about 20-30 MB per hour) and memory hungry. Nanozip uses fewer models for better speed. Options select 1.6 GB memory.

Statistics are stored in hash tables, discarding old data as they fill up.

## Source Code Complexity

To estimate the complexity of a line of code, we compress 29.9 MB of C source code from [gimp](#) v2.0.0 (2004), a graphics editor (from the now defunct UCLC compression benchmark), and header files from [mingw](#) 4.5.0 (2010), a C++ compiler. The code is as follows:

- gimp \*.c, 999 files, 18.180 MB
- gimp \*.h, 775 files, 2.414 MB
- mingw \*.h, 657 files, 9.299 MB

The total is 927,913 lines of C and C++ code with an average length of 32.2 bytes per line. Compressed sizes are as follows:

```
29,893,907 uncompressed
5,066,421 zip -9
3,457,344 7zip -mx
3,433,685 bbb cfm30
2,458,090 nanozip -cc -m1600m
2,450,077 ppmonstr -o32 -m1600
2,113,906 paq8pxd -8
1,919,756 paq8px_v69 -8
1,865,080 paq8pxd_v4 -8
```

The best result is by paq8pxd\_v4, which yields 2.010 bytes or 16.08 bits per line of code.

## Human Genome Complexity

The hg19 human reference genome is a consensus of several anonymous humans. It consists of the following files in FASTA format, with sizes shown:

```
254,235,640 chr1.fa
    108,584 chr1_gl000191_random.fa
    558,468 chr1_gl000192_random.fa
248,063,367 chr2.fa
201,982,885 chr3.fa
194,977,368 chr4.fa
    602,251 chr4_ctg9_hap1.fa
    193,607 chr4_gl000193_random.fa
    195,321 chr4_gl000194_random.fa
184,533,572 chr5.fa
174,537,375 chr6.fa
    4,714,751 chr6_apd_hap1.fa
    4,891,294 chr6_cox_hap2.fa
    4,702,619 chr6_dbb_hap3.fa
```

4,776,945 chr6\_mann\_hap4.fa  
4,930,081 chr6\_mcf\_hap5.fa  
4,704,239 chr6\_qbl\_hap6.fa  
5,027,155 chr6\_ssto\_hap7.fa  
162,321,443 chr7.fa  
    186,576 chr7\_gl000195\_random.fa  
149,291,309 chr8.fa  
    39,715 chr8\_gl000196\_random.fa  
    37,941 chr8\_gl000197\_random.fa  
144,037,706 chr9.fa  
    91,909 chr9\_gl000198\_random.fa  
    173,294 chr9\_gl000199\_random.fa  
    190,798 chr9\_gl000200\_random.fa  
    36,893 chr9\_gl000201\_random.fa  
138,245,449 chr10.fa  
137,706,654 chr11.fa  
    40,929 chr11\_gl000202\_random.fa  
136,528,940 chr12.fa  
117,473,283 chr13.fa  
109,496,538 chr14.fa  
104,582,027 chr15.fa  
    92,161,856 chr16.fa  
    82,819,122 chr17.fa  
    1,714,462 chr17\_ctg5\_hap1.fa  
        38,271 chr17\_gl000203\_random.fa  
        82,960 chr17\_gl000204\_random.fa  
        178,103 chr17\_gl000205\_random.fa  
        41,845 chr17\_gl000206\_random.fa  
79,638,800 chr18.fa  
    4,371 chr18\_gl000207\_random.fa  
60,311,570 chr19.fa  
    94,566 chr19\_gl000208\_random.fa  
    162,376 chr19\_gl000209\_random.fa  
64,286,038 chr20.fa  
49,092,500 chr21.fa  
    28,259 chr21\_gl000210\_random.fa  
52,330,665 chr22.fa  
    16,909 chrM.fa  
    169,914 chrUn\_gl000211.fa  
    190,612 chrUn\_gl000212.fa  
    167,540 chrUn\_gl000213.fa  
    140,489 chrUn\_gl000214.fa  
    176,012 chrUn\_gl000215.fa  
    175,756 chrUn\_gl000216.fa  
    175,608 chrUn\_gl000217.fa  
    164,386 chrUn\_gl000218.fa  
    182,798 chrUn\_gl000219.fa  
    165,055 chrUn\_gl000220.fa  
    158,521 chrUn\_gl000221.fa  
    190,615 chrUn\_gl000222.fa



```
184,081 chrUn_g1000223.fa
183,303 chrUn_g1000224.fa
215,413 chrUn_g1000225.fa
 15,325 chrUn_g1000226.fa
130,958 chrUn_g1000227.fa
131,719 chrUn_g1000228.fa
 20,328 chrUn_g1000229.fa
 44,581 chrUn_g1000230.fa
 27,950 chrUn_g1000231.fa
 41,482 chrUn_g1000232.fa
 46,876 chrUn_g1000233.fa
 41,358 chrUn_g1000234.fa
 35,180 chrUn_g1000235.fa
 42,789 chrUn_g1000236.fa
 46,801 chrUn_g1000237.fa
 40,754 chrUn_g1000238.fa
 34,517 chrUn_g1000239.fa
 42,788 chrUn_g1000240.fa
 43,012 chrUn_g1000241.fa
 44,410 chrUn_g1000242.fa
 44,224 chrUn_g1000243.fa
 40,744 chrUn_g1000244.fa
 37,401 chrUn_g1000245.fa
 38,934 chrUn_g1000246.fa
 37,167 chrUn_g1000247.fa
 40,598 chrUn_g1000248.fa
 39,289 chrUn_g1000249.fa
158,375,978 chrX.fa
 60,561,044 chrY.fa
3,199,905,909 bytes
```

The files are in FASTA format with a one line header like ">chr1" denoting the file name, followed by lines of 50 bases (A,C,G,T,N) terminated by a linefeed. It uses lowercase letters (a,c,g,t) to indicate tandem repeats. It uses N to indicate unknown bases. These usually occur in large blocks around the centromere (about 40% into most of the large files) and smaller blocks scattered throughout the file and at the telomeres on the ends. Out of a total of 3,137,161,264 bases, 239,850,802 (7.6%) are N.

Unreadable bases typically occur in highly repetitive sections of the code. During shotgun sequencing, the chromosome is broken up into small fragments and sequenced in overlapping "reads" of about 100 bases and reassembled. In repetitive regions, there are multiple ways to reassemble the fragments, making them difficult to sequence. The centromere is the "handle" used to pull apart the two copies of the chromosome during mitosis or cell division. The telomeres are trimmed with each replication to prevent runaway growth.

The files *chr1.fa* through *chr22.fa* are the 22 normal chromosomes. Every cell in the body has two of these, one inherited from each parent. *chrX.fa* and *chrY.fa* are the sex chromosomes.

Males have one X and one Y. Females have two X. *chrM.fa* is the mitochondria chromosome, which has its own (slightly different) genetic code. The files ending in *random.fa* are fragments that could not be matched to the main chromosome, so their location is unknown. The files starting with *chrUn* are fragments in which the original chromosome is not known. The files *chr4\_ctg9\_hap1.fa* and the 7 files *chr6\_\*\_hap?.fa* are small regions of chromosomes 4 and 6 (in the middle of the short arm of 6) that too variable between individuals to form a consensus. These nevertheless have a high degree of overlap.

Only about 1.5% of the [human genome](#) consists of exomes, or genes encoding protein. Over half consists of repeating sequences. Some of this serves to regulate genes by binding to proteins that initiate or inhibit transcription. Other sections contain code that is no longer used, or that was inserted by retroviruses and passed on to succeeding generations. Not all of the code is understood.

There are approximately 20,000 genes in the human genome, although the exact number is not known. In contrast, the 1 millimeter long, bacteria eating roundworm, [C. elegans](#) has 20,470 protein encoding genes and another 16,000 RNA encoding genes in only 100M base pairs, 3% of the size of the human genome. If we are to believe that humans are more complex than roundworms, then that complexity must somehow be encoded in the “junk” DNA.

The major source of redundancy in the genome (that we know of) comes from repetitive sequences. There may be many adjacent copies, or they may be widely separated or on different chromosomes. They may be on complementary strands. Only one strand on each chromosome is recorded. The opposite is formed by matching A to T and C to G and reversing the order, for example, *TACT* -> *AGTA*. None of the compressors in our test set are able to recognize complementary strands as contexts. Also, because of the large size of the genome, none of the compressors is able to recognize long distance matches except for *BBB*, and then only if the genome is represented in a more compact form than one base per character.

The obvious way to pack DNA is to use 2 bits per base and 4 bases per byte. However this can make compression worse because two identical strings will appear different to the compressor unless the distance between them is a multiple of 4. To solve this problem, we pack 3 or 4 bases into a byte such that after a while the byte boundaries synchronize. The code we use is the same as the [FASTQZ](#) compressor (Bonfield and Mahoney, 2013). The bases A, T, C, G are encoded as 1, 2, 3, 4 respectively and grouped such that when interpreted in base 4, they form a number in the range 64 to 255. This means that any group starting with G, CG, or CCG is packed 3 to a byte, and all others 4 to a byte. The following example shows how bases would be grouped using different starting points.

```
TGGA ATCA GAT GGA ATCA TCGA ATGG ACTG GAA TGGA ATCA
GGA ATCA GAT GGA ATCA TCGA ATGG ACTG GAA TGGA ATCA
```

```
GAAT CAGA TGGA ATCA TCGA ATGG ACTG GAA TGGA ATCA
AATC AGAT GGA ATCA TCGA ATGG ACTG GAA TGGA ATCA
```

We give higher codes to C and G because they occur less frequently than A and T in the human genome, resulting in tighter packing before compression. Also, we discard all N, under the assumption that the data is highly repetitive and therefore contains very little information. We then compress two ways, once as a single file and once as 26 files. The 26 files are chromosomes 1 through 23, X, Y, M, and Unknown, formed by removing N, concatenating the remaining bases, and packing as described. Variants (chromosomes 4 and 6) and random fragments are concatenated to the chromosomes to which they belong. The unknown fragments go in their own file. For the single file, the compressed sizes (in bytes) are as follows:

```
766,373,649 uncompressed
683,485,287 zip
622,113,887 7zip
605,526,316 bbb
599,775,019 ppmonstr -o8
598,722,820 nanozip
```

As 26 separate files the total compressed sizes are as follows:

```
766,373,636 uncompressed
683,494,823 zip
630,447,231 7zip
628,334,542 bbb
615,908,412 ppmonstr -o8
611,444,955 nanozip
606,201,987 paq8pxd_v4
604,332,601 paq8pxd
```

Compression with *paq8pxd* took 40.6 hours on a 2.0 GHz T3200 processor. The better compression by *paq8pxd\_v4* (a later version) on the source code was due to fixing a problem with overly aggressive file segmentation, which was not a problem with the DNA.

The difference in compressed size for *BBB*, 22,808,226 bytes, is an estimate of the mutual information between chromosomes. The difference is smaller for all other compressors because they could not store the complete statistical model in the 1600 MB of available memory. (*BBB* stores the model in 766 MB of memory and 3 GB of temporary files for the suffix array). This suggests that *paq8pxd\_v1* would have compressed to 581.5 MB given sufficient memory.

To test the effects of including the variants of *chr6*, we compared the compressed sizes (after packing) of *chr6.fa* alone and with the 7 variants concatenated onto the end. The results show that appending the variants only has a very small effect on the total information content,

adding 0.26 MB using the best compressor tested.

<b>chr6 only</b>	<b>plus variants</b>	
44,180,099	51,553,182	packed only
36,762,504	38,590,394	bbb
36,338,669	36,694,017	ppmonstr -o8
36,113,475	36,370,417	nanozip

Although we packed bases with a self-synchronizing code, we nevertheless lose some compression at the beginning of the match before synchronization. To test this effect, we compare compression with and without packing of *chr21* and *chr22*. The unpacked input contains only the letters A, C, G, T, converted to uppercase, discarding the FASTA header, newlines, and all N.

<b>Chr21</b>	<b>Packed</b>	<b>Chr22</b>	<b>Packed</b>	
35,106,642	9,287,838	34,894,545	9,341,723	Uncompressed
7,884,270	7,949,255	7,538,934	7,580,025	bbb
8,134,163	7,802,206	7,853,008	7,377,597	ppmonstr -o8
7,906,238	7,744,100	7,482,284	7,308,297	nanozip

For *nanozip* and *ppmonstr*, packing improves compression because it reduces memory usage and effectively increases the context order. For *BBB*, compression is 0.82% worse for *chr21* and 0.55% for *chr22*. *BBB* (BWT) uses an unbounded context order and is not limited by memory. This suggests that compression could be improved by 2 or 3 MB overall (about 579 MB) by not packing if sufficient memory were available.

Finally, to test the effects of reverse complement contexts, we compute the reverse, the complement, and the reverse complement of *chr21* and *chr22* and append these to the original data (unpacked, but reduced to A, C, G, T as before). The reduction in size of the reverse complement over the other two gives us an estimate of the space that could be saved by recognizing such contexts. Results using *BBB* are as follows:

```
15,768,540 chr21 x 2
15,951,005 chr21 + complement
15,950,060 chr21 + reverse
15,633,621 chr21 + reverse complement

15,077,868 chr22 x 2
15,288,313 chr22 + complement
15,287,298 chr22 + reverse
14,895,566 chr22 + reverse complement
```

Appending the reverse or the complement makes compression worse than storing two compressed copies of the original file. But appending the reverse complement improves compression by 0.86% for *chr21* and 1.20% for *chr22*. This suggests that a 1% (6 MB) improvement might be possible overall, for an information content of 573 MB or  $4.58 \times 10^9$

bits.

There may be many other sources of redundancy that might be discovered with improved compression techniques. That is an area of future work.