LETTER

doi:10.1038/nature11875

Towards practical, high-capacity, low-maintenance information storage in synthesized DNA

Nick Goldman[1], Paul Bertone[1], Siyuan Chen[2], Christophe Dessimoz[1], Emily M. LeProust[2], Botond Sipos[1] & Ewan Birney[1]

Digital production, transmission and storage have revolutionized how we access and use information but have also made archiving an increasingly complex task that requires active, continuing maintenance of digital media. This challenge has focused some interest on DNA as an attractive target for information storage[1] because of its capacity for high-density information encoding, longevity under easily achieved conditions[2–4] and proven track record as an information bearer. Previous DNA-based information storage approaches have encoded only trivial amounts of information[5–7] or were not amenable to scaling-up[8], and used no robust error-correction and lacked examination of their cost-efficiency for large-scale information archival[9]. Here we describe a scalable method that can reliably store more information than has been handled before. We encoded computer files totalling 739 kilobytes of hard-disk storage and with an estimated Shannon information[10] of 5.2×10^6 bits into a DNA code, synthesized this DNA, sequenced it and reconstructed the original files with 100% accuracy. Theoretical analysis indicates that our DNA-based storage scheme could be scaled far beyond current global information volumes and offers a realistic technology for large-scale, long-term and infrequently accessed digital archiving. In fact, current trends in technological advances are reducing DNA synthesis costs at a pace that should make our scheme cost-effective for sub-50-year archiving within a decade.

Although techniques for manipulating, storing and copying large amounts of existing DNA have been established for many years[11–13], one of the main challenges for practical DNA-based information storage is the difficulty of synthesizing long sequences of DNA *de novo* to an exactly specified design. As in the approach of ref. 9, we represent the information being stored as a hypothetical long DNA molecule and encode this *in vitro* using shorter DNA fragments. This offers the benefits that isolated DNA fragments are easily manipulated *in vitro*[11,13], and that the routine recovery of intact fragments from samples that are tens of thousands of years old[14,15] indicates that well-prepared synthetic DNA should have an exceptionally long lifespan in low-maintenance environments[3,4]. In contrast, approaches using living vectors[6–8] are not as reliable, scalable or cost-efficient owing to disadvantages such as constraints on the genomic elements and locations that can be manipulated without affecting viability, the fact that mutation will cause the fidelity of stored and decoded information to reduce over time, and possibly the requirement for storage conditions to be carefully regulated. Existing schemes used for DNA computing in principle permit large-scale memory[1,16], but data encoding in DNA computing is inextricably linked to the specific application or algorithm[17] and no practical storage schemes have been realized.

As a proof of concept for practical DNA-based storage, we selected and encoded a range of common computer file formats to emphasize the ability to store arbitrary digital information. The five files comprised all 154 of Shakespeare's sonnets (ASCII text), a classic scientific paper[18] (PDF format), a medium-resolution colour photograph of the European Bioinformatics Institute (JPEG 2000 format), a 26-s excerpt from Martin Luther King's 1963 'I have a dream' speech (MP3 format) and a Huffman code[10] used in this study to convert bytes to base-3 digits (ASCII text), giving a total of 757,051 bytes or a Shannon information[10] of 5.2×10^6 bits (see Supplementary Information and Supplementary Table 1 for full details).

The bytes comprising each file were represented as single DNA sequences with no homopolymers (runs of $\geq$2 identical bases, which are associated with higher error rates in existing high-throughput sequencing technologies[19] and led to errors in a recent DNA-storage experiment[9]). Each DNA sequence was split into overlapping segments, generating fourfold redundancy, and alternate segments were converted to their reverse complement (see Fig. 1 and Supplementary Information). These measures reduce the probability of systematic failure for any particular string, which could lead to uncorrectable errors and data loss. Each segment was then augmented with indexing information that permitted determination of the file from which it originated and its location within that file, and simple parity-check error-detection[10]. In all, the five files were represented by a total of 153,335 strings of DNA, each comprising 117 nucleotides (nt). The perfectly uniform fragment lengths and absence of homopolymers make it obvious that the synthesized DNA does not have a natural (biological) origin, and so imply the presence of deliberate design and encoded information[2].

We synthesized oligonucleotides (oligos) corresponding to our designed DNA strings using an updated version of Agilent Technologies' OLS (oligo library synthesis) process[20], creating $\sim 1.2 \times 10^7$ copies of each DNA string. Errors occur only rarely ($\sim$1 error per 500 bases) and independently in the different copies of each string, again enhancing our method's error tolerance. We shipped the synthesized DNA in lyophilized form that is expected to have excellent long-term preservation characteristics[3,4], at ambient temperature and without specialized packaging, from the USA to Germany via the UK. After resuspension, amplification and purification, we sequenced a sample of the resulting library products at the EMBL Genomics Core Facility in paired-end mode on the Illumina HiSeq 2000. We transferred the remainder of the library to multiple aliquots and re-lyophilized these for long-term storage.

Our base calling using AYB[21] yielded 79.6×10^6 read-pairs of 104 bases in length, from which we reconstructed full-length (117-nt) DNA strings *in silico*. Strings with uncertainties due to synthesis or sequencing errors were discarded and the remainder decoded using the reverse of the encoding procedure, with the error-detection bases and properties of the coding scheme allowing us to discard further strings containing errors. Although many discarded strings will have contained information that could have been recovered with more sophisticated decoding, the high level of redundancy and sequencing coverage rendered this unnecessary in our experiment. Full-length DNA sequences representing the original encoded files were then reconstructed *in silico*. The decoding process used no additional information derived from knowledge of the experimental design. Full details of the encoding, sequencing and decoding processes are given in Supplementary Information.

Four of the five resulting DNA sequences could be fully decoded without intervention. The fifth however contained two gaps, each a run

[1]European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton CB10 1SD, UK. [2]Agilent Technologies, Genomics–LSSU, 5301 Stevens Creek Boulevard, Santa Clara, California 95051, USA.

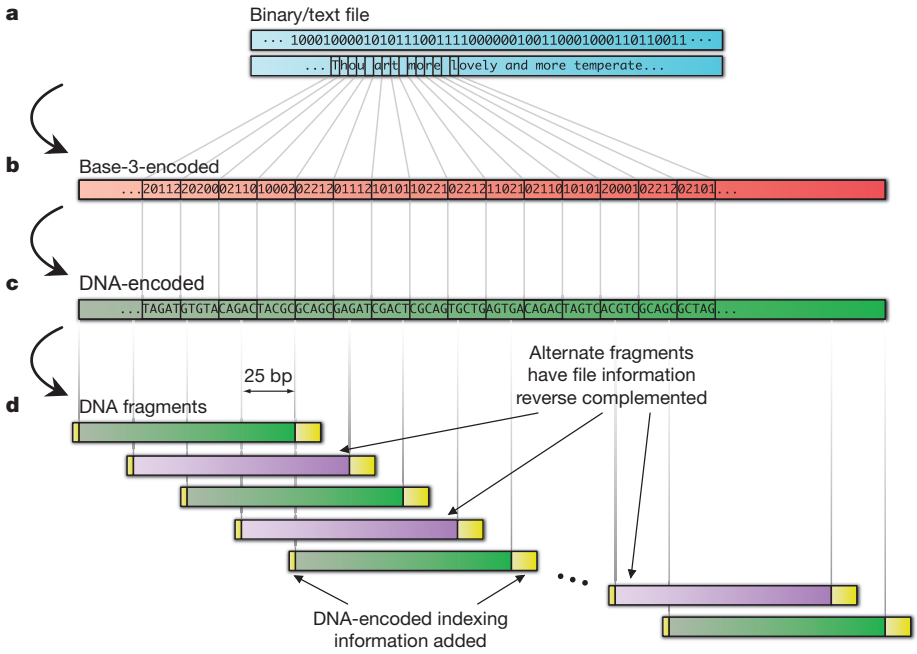


Figure 1 | Digital information encoding in DNA. Digital information (**a**, in blue), here binary digits holding the ASCII codes for part of Shakespeare's sonnet 18, was converted to base-3 (**b**, red) using a Huffman code that replaces each byte with five or six base-3 digits (trits). This in turn was converted *in silico* to our DNA code (**c**, green) by replacement of each trit with one of the three nucleotides different from the previous one used, ensuring no homopolymers were generated. This formed the basis for a large number of overlapping segments of length 100 bases with overlap of 75 bases, creating fourfold redundancy (**d**, green and, with alternate segments reverse complemented for added data security, violet). Indexing DNA codes were added (yellow), also encoded as non-repeating DNA nucleotides. See Supplementary Information for further details.

of 25 bases, for which no segment was detected corresponding to the original DNA. Each of these gaps was caused by the failure to sequence any oligo representing any of four consecutive overlapping segments. Inspection of the neighbouring regions of the reconstructed sequence permitted us to hypothesize what the missing nucleotides should have been (see Supplementary Information) and we manually inserted those 50 bases accordingly. This sequence could also then be decoded. Inspection confirmed that our original computer files had been reconstructed with 100% accuracy.

An important issue for long-term digital archiving is how DNA-based storage scales to larger applications. The number of bases of synthesized DNA needed to encode information grows linearly with the amount of information to be stored, but we must also consider the indexing information required to reconstruct full-length files from short fragments. As indexing information grows only as the logarithm of the number of fragments to be indexed, the total amount of synthesized DNA required grows sub-linearly. Increasingly large parts of each fragment are needed for indexing however and, although it is reasonable to expect synthesis of longer strings to be possible in future, we modelled the behaviour of our scheme under the conservative constraint of a constant 114 nt available for both data and indexing information (see Supplementary Information). As the total amount of information increases, the encoding efficiency decreases only slowly (Fig. 2a). In our experiment (megabyte scale) the encoding scheme is 88% efficient; Fig. 2a indicates that efficiency remains >70% for data storage on petabyte (PB, 10^{15} bytes) scales and >65% on exabyte (EB, 10^{18} bytes) scales, and that DNA-based storage remains feasible on scales many orders of magnitude greater than current global data volumes[22]. Figure 2a also shows that costs (per unit information stored) rise only slowly as data volumes increase over many orders of magnitude. Efficiency and costs scale even more favourably if we consider the synthesized fragment lengths available using the latest technology (Supplementary Fig. 5).

As the amount of information stored increases, decoding requires more strings to be sequenced. A fixed decoding expenditure per byte of encoded information would mean that each base is read fewer times and so is more likely to suffer decoding error. But extension of our scaling analysis to model the influence of reduced sequencing coverage on the per-decoded-base error rate (see Supplementary Information) revealed that error rates increase only very slowly as the amount of information encoded increases to a global data scale and beyond (Supplementary Table 4). This also suggests that our mean sequencing coverage of 1,308 times was considerably in excess of that needed for reliable decoding. We confirmed this by subsampling from the 79.6×10^6 read-pairs to simulate experiments with lower coverage. Figure 2b indicates that reducing the coverage by a factor of 10 (or even more) would have led to unaltered decoding characteristics, which further illustrates the robustness of our DNA-storage method.

DNA-based storage might already be economically viable for long-horizon archives with a low expectation of extensive access, such as government and historical records[23,24]. An example in a scientific context is CERN's CASTOR system[25], which stores a total of 80 PB of Large Hadron Collider data and grows at 15 PB yr^{-1}. Only 10% is maintained on disk, and CASTOR migrates regularly between magnetic tape formats. Archives of older data are needed for potential future verification of events, but access rates decrease considerably 2–3 years after collection. Further examples are found in astronomy, medicine and interplanetary exploration[26]. With negligible computational costs and optimized use of the technologies we employed, we estimate current costs to be \$12,400 MB^{-1} for information storage in DNA and \$220 MB^{-1} for information decoding. Modelling relative long-term costs of archiving using DNA-based storage or magnetic tape shows that the key parameters are the ratio of the one-time cost of synthesizing the DNA to the recurrent fixed cost of transferring data between tape technologies or media, which we estimate to be 125–500 currently, and the frequency of tape transition events (Supplementary Information and Supplementary Fig. 7). We find that with current technology and our encoding scheme, DNA-based storage may be cost-effective for archives of several megabytes with a ~600–5,000-yr horizon (Fig. 2c). One order of magnitude reduction in synthesis costs reduces this to ~50–500 yr; with two orders

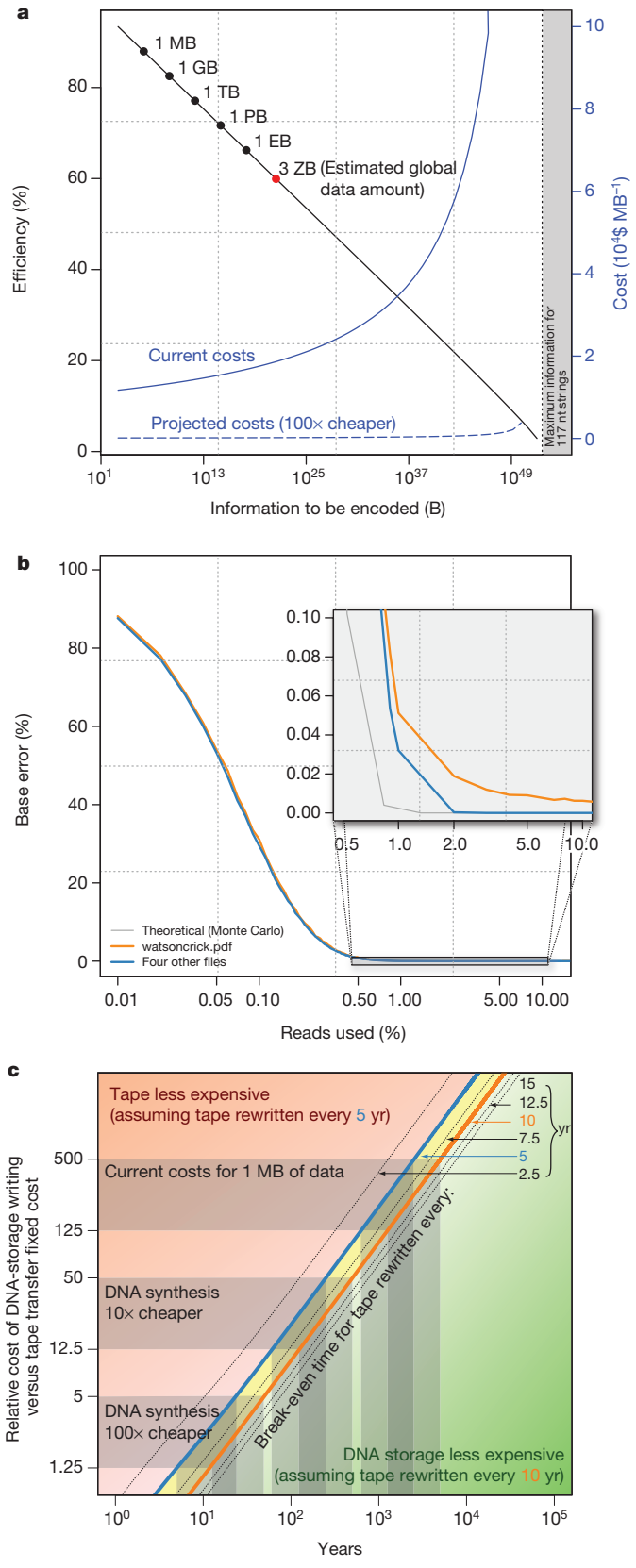


Figure 2 | Scaling properties and robustness of DNA-based storage. **a**, Encoding efficiency and costs change as the amount of stored information increases. The *x* axis (logarithmic scale) represents the total amount of information to be encoded. Common data scales are indicated, including the three zettabyte (3 ZB, 3×10^{21} bytes) global data estimate, shown red. The black line (*y*-axis scale to left) indicates encoding efficiency, measured as the proportion of synthesized bases available for data encoding. The blue curves (*y*-axis scale to right) indicate the corresponding effect on encoding costs, both at current synthesis cost levels (solid line) and in the case of a two-order-of-magnitude reduction (dashed line). **b**, Per-recovered-base error rate (*y* axis) as a function of sequencing coverage, represented by the percentage of the original 79.6×10^6 read-pairs sampled (*x* axis; logarithmic scale). The blue curve represents the four files recovered without human intervention: the error is zero when ≥2% of the original reads are used. The grey curve is obtained by Monte Carlo simulation from our theoretical error rate model. The orange curve represents the file (watsoncrick.pdf) that required manual correction: the minimum possible error rate is 0.0036%. The boxed area is shown magnified in the inset. **c**, Timescales for which DNA-based storage is cost-effective. The blue curve indicates the relationship between break-even time beyond which DNA storage is less expensive than magnetic tape (*x* axis) and relative cost of DNA-storage synthesis and tape transfer fixed costs (*y* axis), assuming the tape archive has to be read and rewritten every 5 yr. The orange curve corresponds to tape transfers every 10 yr; broken curves correspond to other transfer periods as indicated. In the green-shaded region, DNA storage is cost-effective when transfers occur more frequently than every 10 yr; in the yellow-shaded region, DNA storage is cost-effective when transfers occur every 5–10 yr; in the red-shaded region tape is less expensive when transfers occur less frequently than every 5 yr. Grey-shaded ranges of relative costs of DNA synthesis to tape transfer are 125–500 (current costs for 1 MB of data), 12.5–50 (achieved if DNA synthesis costs are reduced by one order of magnitude) and 1.25–5 (costs reduced by two orders of magnitude). Note the logarithmic scales on both axes. See Supplementary Information for further details.

of magnitude reduction, as can be expected in less than a decade if current trends continue (ref. 13, and http://www.synthesis.cc/2011/06/new-cost-curves.html), DNA-based storage becomes practical for archives with a horizon of less than 50 yr. The speed of DNA-storage writing and reading are not competitive with current technology, but both processes can be accelerated through parallelization (Supplementary Information).

The DNA-based storage medium has different properties from traditional tape- or disk-based storage. As DNA is the basis of life on Earth, methods for manipulating, storing and reading it will remain the subject of continual technological innovation. As with any storage system, a large-scale DNA archive would need stable DNA management[27] and physical indexing of depositions. But whereas current digital schemes for archiving require active and continuing maintenance and regular transferring between storage media, the DNA-based storage medium requires no active maintenance other than a cold, dry and dark environment[3,4] (such as the Global Crop Diversity Trust's Svalbard Global Seed Vault, which has no permanent on-site staff[28]) yet remains viable for thousands of years even by conservative estimates. We achieved an information storage density of ~2.2 PB g^{-1} (Supplementary Information). Our sequencing protocol consumed just 10% of the library produced from the synthesized DNA (Supplementary Table 2), already leaving enough for multiple equivalent copies. Existing technologies for copying DNA are highly efficient[11,13], meaning that DNA is an excellent medium for the creation of copies of any archive for transportation, sharing or security. Overall, DNA-based storage has potential as a practical solution to the digital archiving problem and may become a cost-effective solution for rarely accessed archives.

Received 15 May; accepted 12 December 2012.
Published online 23 January 2013.

1. Baum, E. B. Building an associative memory vastly larger than the brain. *Science* **268,** 583–585 (1995).
2. Cox, J. P. L. Long-term data storage in DNA. *Trends Biotechnol.* **19,** 247–250 (2001).
3. Anchordoquy, T. J. & Molina, M. C. Preservation of DNA. *Cell Preserv. Technol.* **5,** 180–188 (2007).
4. Bonnet, J. *et al.* Chain and conformation stability of solid-state DNA: implications for room temperature storage. *Nucleic Acids Res.* **38,** 1531–1546 (2010).
5. Clelland, C. T., Risca, V. & Bancroft, C. Hiding messages in DNA microdots. *Nature* **399,** 533–534 (1999).
6. Kac, E. *Genesis* (1999); available at http://www.ekac.org/geninfo.html (accessed 10 May 2012).

7. Ailenberg, M. & Rotstein, O. D. An improved Huffman coding method for archiving text, images, and music characters in DNA. *Biotechniques* **47,** 747–754 (2009).
8. Gibson, D. G. *et al.* Creation of a bacterial cell controlled by a chemically synthesized genome. *Science* **329,** 52–56 (2010).
9. Church, G. M., Gao, Y. & Kosuri, S. Next-generation digital information storage in DNA. *Science* **337,** 1628 (2012).
10. MacKay, D. J. C. *Information Theory, Inference, and Learning Algorithms* (Cambridge Univ. Press, 2003).
11. Erlich, H. A., Gelfand, D. & Sninsky, J. J. Recent advances in the polymerase chain reaction. *Science* **252,** 1643–1651 (1991).
12. Monaco, A. P. & Larin, Z. YACs, BACs, PACs and MACs: artificial chromosomes as research tools. *Trends Biotechnol.* **12,** 280–286 (1994).
13. Carr, P. A. & Church, G. M. Genome engineering. *Nature Biotechnol.* **27,** 1151–1162 (2009).
14. Willerslev, E. *et al.* Ancient biomolecules from deep ice cores reveal a forested southern Greenland. *Science* **317,** 111–114 (2007).
15. Green, R. E. *et al.* A draft sequence of the Neandertal genome. *Science* **328,** 710–722 (2010).
16. Kari, L. & Mahalingam, K. in *Algorithms and Theory of Computation Handbook* Vol. 2, 2nd edn (eds Atallah, M. J. & Blanton, M.) 31-1–31-24 (Chapman & Hall, 2009).
17. Păun, G., Rozenberg, G. & Salomaa, A. *DNA Computing: New Computing Paradigms* (Springer, 1998).
18. Watson, J. D. & Crick, F. H. C. Molecular structure of nucleic acids. *Nature* **171,** 737–738 (1953).
19. Niedringhaus, T. P., Milanova, D., Kerby, M. B., Snyder, M. P. & Barron, A. E. Landscape of next-generation sequencing technologies. *Anal. Chem.* **83,** 4327–4341 (2011).
20. LeProust, E. M. *et al.* Synthesis of high-quality libraries of long (150mer) oligonucleotides by a novel depurination controlled process. *Nucleic Acids Res.* **38,** 2522–2540 (2010).
21. Massingham, T. & Goldman, N. All Your Base: a fast and accurate probabilistic approach to base calling. *Genome Biol.* **13,** R13 (2012).
22. Gantz, J. & Reinsel, D. *Extracting Value from Chaos* (IDC, 2011).
23. Brand, S. *The Clock of the Long Now* (Basic Books, 1999).
24. Digital archiving. History flushed. *Economist* **403,** 56–57 (28 April 2012); available at http://www.economist.com/node/21553410 (2012).
25. Bessone, N., Cancio, G., Murray, S. & Taurelli, G. Increasing the efficiency of tape-based storage backends. *J. Phys. Conf. Ser.* **219,** 062038 (2010).
26. Baker, M. *et al.* in *Proc. 1st ACM SIGOPS/EuroSys European Conf. on Computer Systems* (eds Berbers, Y. & Zwaenepoel, W.) 221–234 (ACM, 2006).
27. Yuille, M. *et al.* The UK DNA banking network: a ''fair access'' biobank. *Cell Tissue Bank.* **11,** 241–251 (2010).
28. Global Crop Diversity Trust. Svalbard Global Seed Vault. (2012); available at http://www.croptrust.org/main/content/svalbard-global-seed-vault (accessed 10 May 2012).

Supplementary Information is available in the online version of the paper.

Acknowledgements At the University of Cambridge: D. MacKay and G. Mitchison for advice on codes for run-length-limited channels. At CERN: B. Jones for discussions on data archival. At EBI: A. Löytynoja for custom multiple sequence alignment software, H. Marsden for computing base calls and for detecting an error in the original parity-check encoding, T. Massingham for computing base calls and advice on code theory and K. Gori, D. Henk, R. Loos, S. Parks and R. Schwarz for assistance with revisions to the manuscript. In the Genomics Core Facility at EMBL Heidelberg: V. Benes for advice on Next-Generation Sequencing protocols, D. Pavlinić for sequencing and J. Blake for data handling. C.D. is supported by a fellowship from the Swiss National Science Foundation (grant 136461). B.S. is supported by an EMBL Interdisciplinary Postdoctoral Fellowship under Marie Curie Actions (COFUND).

Author Contributions N.G. and E.B. conceived and planned the project and devised the information-encoding methods. P.B. advised on oligo design and Next-Generation Sequencing protocols, prepared the DNA library and managed the sequencing process. S.C. and E.M.L. provided custom oligonucleotides. N.G. wrote the software for encoding and decoding information into/from DNA and analysed the data. N.G., E.B., C.D. and B.S. modelled the scaling properties of DNA storage. N.G. wrote the paper with discussions and contributions from all other authors. N.G. and C.D. produced the figures.

Author Information Data are available at http://www.ebi.ac.uk/goldman-srv/DNA-storage and in the Sequence Read Archive (SRA) with accession number ERP002040. Reprints and permissions information is available at www.nature.com/reprints. The authors declare competing financial interests: details are available in the online version of the paper. Readers are welcome to comment on the online version of the paper. Correspondence and requests for materials should be addressed to N.G. (goldman@ebi.ac.uk).

doi:10.1038/nature11875

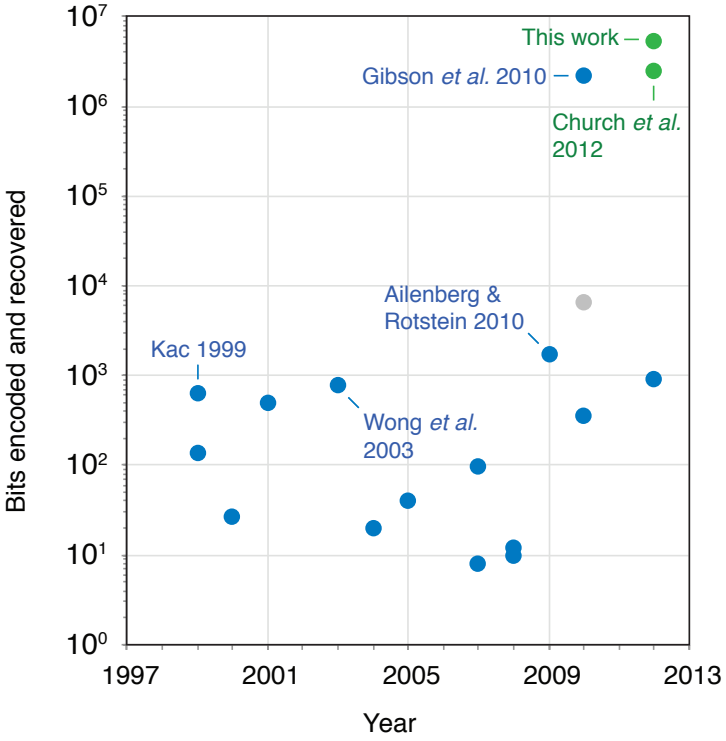
HUMAN-DESIGNED INFORMATION STORED IN DNA

Supplementary Table 1 and Supplementary Fig. 1 show the amounts of human-designed information stored in DNA and successfully recovered in this letter and 16 previous studies. Supplementary Fig. 2 illustrates some of the information encoded in this study. The Shannon information content[10] of the designed messages was approximated by the minimum number of bits required to encode the message using any of the following methods:

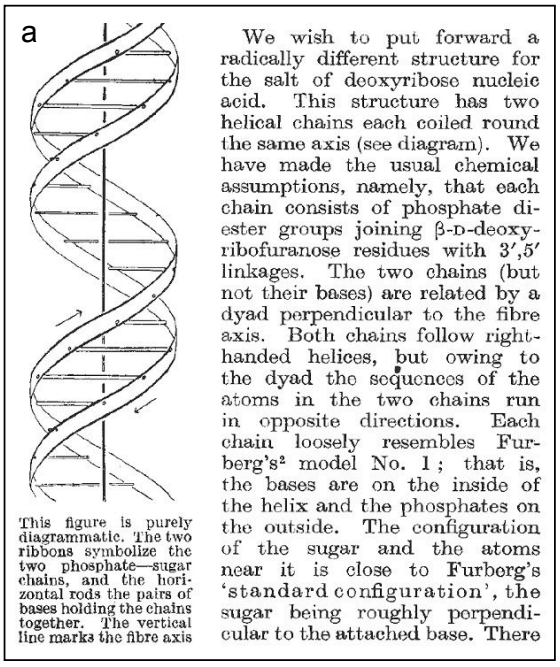
- compress ASCII file containing the message in natural form, using Unix command `gzip --best`
- compress ASCII file containing the message in natural form, using Unix command `bzip2 --best`
- for DNA sequence, 2 bits per base
- for simple English text, 5 bits per character (permits use of $2^5 = 32$ characters, e.g. 26 letters of the alphabet plus space and simple punctuation)
- for English/Latin text using reduced or extended alphabets, the number of bits per character is calculated similarly (e.g. 3 bits per character for an alphabet of $8 = 2^3$ characters, 6 bits/char for a 64-character alphabet)

Ref.	Authors	Year	Message type	Message length	Bases used	Shannon information (bits)	Notes
5	Clelland et al.	1999	English text	23 characters	69	138	
6	Kac	1999	English text	129 characters	360	645	Biblical quotation encoded in mutating *E. coli* genome as a work of art; decoded with 3 character errors, attributed to mutation
29	Leier et al.	2000	three 9-bit numbers	27 bits	810	27	
30	Bancroft et al.	2001	English text	106 characters	318	504	
31	Wong et al.	2003	English text (64 character alphabet)	185 characters*	560	800	*estimated
32	Arita & Ohashi	2004	English text	4 characters	24	20	
33	Kashimawura et al.	2005	DNA string	20 bases	65	40	
34	Skinner et al.	2007	four 2-bit numbers	8 bits	231	8	
35	Yachie et al.	2007	mathematical equation and date (256 character alphabet)	12 characters	250	96	
36	Heider & Barnekow	2008	English text	2 characters	5	10	
37	Portney et al.	2008	Latin text (using 8 character alphabet)	5 characters	80	12	final character of 5 (i.e. 3 bits) lost in decoding
7	Ailenberg & Rotstein	2009	English text, simple musical notation, simple line-drawing notation	349 characters	844	1715	
38	CUHK-iGEM	2010	English text	70 characters	438	350	
8	Gibson et al.	2010	bacterial genome with additional "watermark" sequences (see below)	1077947 bases	1077947	2155894	decoded with 8 base errors and two insertions of 768 and 85 bases, respectively
	of which:		"watermarks": English text plus programming symbols (64 character alphabet)	1280 characters	4658	6504	
9	Church et al.	2012	English text, JPEG images, computer code, all within HTML encoding	658776 characters (bytes)	6313270	2495760	decoded with 10 bit errors
39	Jarvis & NPC	2012	English text	180 characters	540	900	Article 1 of the Universal Declaration of Human Rights encoded in *E. coli* genome as work of art
	Goldman et al.	**this letter**	**Total**	**757051 bytes**	**17940195**	**5165800**	
	of which:		English text (all 154 Shakespeare sonnets)	107738 characters (bytes)	2533635	297856	file `wssnt10.txt` (from Project Gutenberg, `http://www.gutenberg.org/ebooks/1041`)
			PDF document (Watson and Crick, 1953)	280864 bytes	6659172	2119848	file `watsoncrick.pdf` (from the Nature website, `http://www.nature.com/nature/dna50/archive.html`, modified to achieve higher compression); see Supplementary Fig. 2a
			MP3 audio file (extract from Martin Luther King "I Have a Dream" speech)	168539 bytes	3997773	1227176	file `MLK_excerpt_VBR_45-85.mp3` (from `http://www.americanrhetoric.com/speeches/mlkihaveadream.htm`, modified to achieve higher compression: variable bit rate, typically 48–56 kbps; sampling frequency 44.1 kHz)
			JPEG 2000 image file (image of EBI, 640 x 480 pixels, 16.7M colours)	184264 bytes	4379076	1474000	file `EBI.jp2` (authors' own picture); see Supplementary Fig. 2b
			ASCII file (Huffman code used to convert bytes to base-3; human readable)	15646 bytes	370539	46920	file `View_huff3.cd.new`

Supplementary Table 1 | Amounts of human-designed information stored in DNA and successfully recovered. Message length uses the natural measurement according to the Message type. Bases used indicates the number of DNA bases designed to contain a single copy of the encoded message and ignores the number of copies synthesized.



Supplementary Figure 1 | Amounts of human-designed information stored in DNA and successfully recovered. Information content is measured in bits; note the logarithmic scale on the *y*-axis. Blue points indicate studies not adapted to high-throughput data storage; green indicates high-throughput methods. The grey point indicates that part of the Gibson *et al.* (2010) experiment[8] that encoded information of non-biological origin.


a

This figure is purely diagrammatic. The two ribbons symbolize the two phosphate—sugar chains, and the horizontal rods the pairs of bases holding the chains together. The vertical line marks the fibre axis

We wish to put forward a radically different structure for the salt of deoxyribose nucleic acid. This structure has two helical chains each coiled round the same axis (see diagram). We have made the usual chemical assumptions, namely, that each chain consists of phosphate diester groups joining β-D-deoxyribofuranose residues with 3′,5′ linkages. The two chains (but not their bases) are related by a dyad perpendicular to the fibre axis. Both chains follow right-handed helices, but owing to the dyad the sequences of the atoms in the two chains run in opposite directions. Each chain loosely resembles Furberg's[2] model No. 1; that is, the bases are on the inside of the helix and the phosphates on the outside. The configuration of the sugar and the atoms near it is close to Furberg's 'standard configuration', the sugar being roughly perpendicular to the attached base. There


b

Supplementary Figure 2 | Digital information encoded in DNA. a, An excerpt from the Watson and Crick (1953) paper[18] (PDF format) and **b**, a digital photograph of the European Bioinformatics Institute (JPEG 2000 format) that were among the files encoded in DNA and successfully recovered in this study.

SUPPLEMENTARY METHODS

Digital information encoding. Five files of digital information stored on a hard disk drive were encoded using purpose-written computer software. Each byte of each file to be encoded was represented as a sequence of DNA bases via base-3 digits ('trits' 0, 1 and 2) using a purpose-designed Huffman code[10]. Each of the 256 possible bytes was represented by five or six trits; the Huffman code is given in Supplementary File `huffman.pdf`. Next, each trit was encoded as a DNA nucleotide selected from the three nucleotides different from the last one used, to exclude homopolymer runs. The resulting DNA sequence was converted to segments of length 100 bases, each overlapping the previous by 75 bases, to give strings of a length that was readily synthesized and to provide fourfold redundancy (each DNA base is included in four different segments). Alternate segments were reverse complemented. Indexing information, comprising two trits for file identification (permitting up to $3^2 = 9$ files to be distinguished, in this implementation), 12 trits for intra-file location information (permitting up to $3^{12} = 531{,}441$ locations per file, i.e. a total of up to $3^{14} = 4{,}782{,}969$ unique data locations) and one parity-check[10] trit, again encoded as non-repeating DNA nucleotides, was appended to the 100 information storage bases. Each indexed DNA segment had one further base added to each end, consistent with the 'no homopolymers' rule, that would indicate whether the entire fragment was reverse complemented during the 'reading' stage of the experiment. A full formal specification of the digital information encoding scheme is given in Supplementary File `file2features.pdf`. In total, the five files were represented by a total of 153,335 strings of DNA, each comprising 117 nt (= 1 + 100 + 2 + 12 + 1 + 1) to encode original digital information plus indexing information. The fourfold redundancy provides simple but effective error correction: as each base is encoded in four of the DNA segments, two of which are reverse complemented, any systematic or chance errors in synthesis or sequencing may be corrected by majority vote or more complex decoding schemes. We used simple majority voting (see below).

The data-encoding component of each string can contain Shannon information at 5.07 DNA bases per byte (i.e. 8/5.07 = 1.58 bits per base), close to the theoretical optimum capacity of 5.05 bits per DNA base (see `huffman.pdf`) for base-4 channels with runlength limited to 1 (i.e. no repeated nucleotides). After error-correction redundancy and addition of indexing and parity-check information, the data content of our encoding scheme was 4.94 bytes per string (= 757,051/153,335), or 0.0422 bytes per base (= 4.94/117) (i.e. 23.70 bases per byte). The 153,335 designed DNA strings are available online at http://www.ebi.ac.uk/goldman-srv/DNA-storage.

Our indexing scheme, with 14 nt per string available to record file identification and intra-file location, is easily extended by the addition of further indexing nucleotides. This is considered below, in our analysis of the scaling properties of our DNA-storage scheme. Increasing the number of indexing trits (and therefore bases) used to specify file and intra-file location by just two, to 16, gives $3^{16} = 43{,}046{,}721$ unique locations, in excess of the 16.8M that is the practical maximum for the Nested Primer Molecular Memory (NPMM) scheme[40,16]. While these indexing schemes share the aim of encoding which part of a larger total message any one string contains, ours is simpler and more-readily-extensible as it does not incorporate any system by which the indexing information is used to physically extract a subset of the information-bearing strings prior to decoding, as the NPMM scheme does.

DNA synthesis. The synthesis process was also used to incorporate 33 nt paired-end adapter sequences at the 5′ and 3′ ends of each oligonucleotide (oligo) to facilitate PCR amplification and sequencing on the Illumina platform:

- 5′ adapter: `5′-ACACTCTTTCCCTACACGACGCTCTTCCGATCT-3′`
- 3′ adapter: `5′-AGATCGGAAGAGCGGTTCAGCAGGAATGCCGAG-3′`

The 153,335 DNA oligo designs were synthesized in three distinct runs (with oligos randomly assigned to runs) using an updated version of Agilent Technologies' OLS (oligo library synthesis) process described previously[41,20]. This adapts the phosphoramidite chemistry developed previously[42] and employs inkjet printing and flowcell reactor technologies in the SurePrint *in situ* microarray synthesis platform. Inkjet printing within an anhydrous chamber allows the delivery of very small volumes of phosphoramidites to a confined coupling area on a 2D planar surface, resulting in the addition of hundreds of thousands of bases in parallel. Subsequent oxidation and detritylation are carried out in a flowcell reactor. Once DNA synthesis has been completed, the oligos are then cleaved from the surface and deprotected[43].

Up to ~99.8% coupling efficiency is achieved by using thousands-fold excess of phosphoramidite and activator solution. Similarly, millions-fold excess of detritylation agent drives the removal of the 5′-hydroxyl protecting group to near-completion. A novel controlled process in the flowcell reactor significantly reduces depurination, the most prevalent side reaction[20]. With the latest platform, up to 244,000 unique sequences are synthesized in parallel and delivered as ~1–10 pmol pools of oligos. This is equivalent to ~2.5–25 × 10^6 oligos for each designed sequence (= 1–10 × 10^{-12} × 6.02 × 10^{23}/244,000). In our experiment, three runs were used to synthesize 153,335 designs, leading to the higher figure of ~12–120 × 10^6 (= 3–30 × 10^{-12} × 6.02 × 10^{23}/153,335).

Error rates in the Agilent OLS process are approximately 1 per 500 bases synthesized[41] and synthesis errors are believed to occur independently in different oligos (SC and EML, unpublished data). Combined with our data encoding scheme, this gives further error tolerance. The probability that a given oligo is synthesized entirely correctly is ~0.79 (= $(1 - 1/500)^{117}$), giving a large pool of correct oligos; oligos with a small number of errors in their 100 nt data region may also contribute to correct decoding, with the majority of positions contributing correct information and a small number of errors being outweighed by contributions from other reads (see below).

Library preparation and sequencing. The three samples of lyophilised oligos were resuspended in Tris buffer to a concentration of 5 ng/ml. Samples were then purified from residual synthesis by-products on Ampure XP paramagnetic beads (Beckman Coulter). The reconstituted oligo library was amplified in a total of 22 cycles using thermocycler conditions selected for even A/T vs. G/C processing[44]. PCR was performed with high-fidelity AccuPrime reagents (Invitrogen), a combination of Taq and *pyrococcus* polymerases with a thermostable accessory protein, and paired-end PCR primers (Illumina) complementary to the synthesized adapter sequences flanking each DNA-storage oligo to incorporate additional sequences necessary for flowcell attachment. PCR amplification enabled enrichment for full-length oligos with both 5′ and 3′ adapters correctly synthesized, and allowed us to achieve appropriate concentration for sequencing while simultaneously incorporating the additional sequences necessary for flowcell attachment and cluster formation. The amplified library products were bead-purified and quantified on the Agilent 2100 Bioanalyzer (concentration

determined to be 15.1 ng/µl, i.e. 86 nM given a peak construct size measured at 270 bp and approximating 650 pg/pmol per bp), diluted to a concentration of 16 pM for flowcell loading and sequenced in paired-end mode on the Illumina HiSeq 2000. The sequencing reaction consumed ~0.1% of the DNA in the initial library: 337 pg of DNA (120µl at 16 pM) from a starting value of 302 ng (20µl at 86 nM). Further details of the sample preparation are given in Supplementary Table 2.

Sample stage	Vol (µl)	Conc (ng/µl)	Conc (nM)	Amount of DNA (pg)	Description
A	20	15.1	86	302000	PCR products, quantified on the Agilent 2100 Bioanalyzer
B	2	15.1	86	30200	2 µl of (A) extracted for sequencing; remaining 18 µl reserved for future analysis
C	17.2	1.76	10	30200	(B) then diluted in Tris to 10 nM concentration
D	2	1.76	10	3512	2 µl of (C) retained (remainder discarded in this experiment)
E	20	0.176	1	3512	(D) then denatured in 100mM NaOH, giving dilution to 1 nM
F	16	0.176	1	2809	16 µl of (E) retained (remainder discarded in this experiment)
G	1000	0.00281	0.016	2809	(F) then diluted to 1000 µl
H	120	0.00281	0.016	337	120 µl of (G) sent for clustering (remainder discarded in this experiment)

Supplementary Table 2 | Sample preparation details. Products at stage A (blue) were measured; other values were computed from these.

Base calls were computed from observed intensities using the AYB software[21], producing 79.6M read-pairs of 104 bases in length. (Illumina's base calling software Bustard produced 65.9M read-pairs, 17.2% fewer than AYB, but led to qualitatively identical decoding results.) Quality control of the reads was performed using FastQC (version 0.10.1; ref. 45). Overall the QC report (available as Supplementary File `FastQC.pdf`) indicated a high-quality sequencing run. Per-cycle quality scores were as expected for an Illumina HiSeq run. The mean quality (Q) score was 36.7, with 95% of quality scores ≥ Q30. The GC content of the sequenced reads and the *k*-mer frequencies along the reads were consistent with the structure of the designed DNA strings. The read duplication levels were high, in concordance with the design of the library providing many reads covering any single string.

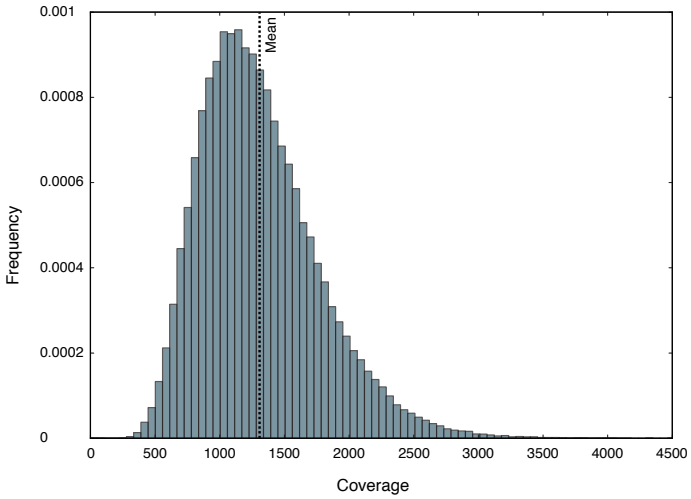
As further quality assessment, but not used for subsequent digital information decoding, the reads were aligned to the designed DNA strings using BWA version 0.6.1-r104 in paired-end mode[46]. Per-cycle error rates were calculated from the resulting alignment using the ErrorRatePerCycle functionality of the GATK package (version 2.1-8-g5efb575; ref. 47). Per-cycle error rates were as expected for an Illumina HiSeq run (see Supplementary File `GATK.txt`) and the mean error rate of 0.001774 after 12.81% unmappable reads were discarded is in line with the combination of current estimates of synthesis error (1 base in 500, above) and sequencing error (1 base in 1,000; ref. 48).

Digital information decoding. As the central 91 bases of each oligo were sequenced from both ends, rapid computation of full-length (117 base) oligos and removal of reads

inconsistent with our designs was straightforward. Sequencing reads were decoded using purpose-written software that exactly reverses the encoding process. The numbers of reads used in different stages of the information decoding process are given in Supplementary Table 3. At the final stage of decoding, the five files were reconstructed from 50.1M strings, giving a mean sequencing depth of 1,308× coverage (standard deviation 459). Supplementary Fig. 3 shows the distribution of sequencing depths over encoded data locations (bases of the files' DNA representations). Virtually every location within each decoded file was detected in hundreds or thousands of different sequenced DNA oligos.

Analysis stage	Number of reads	% of total	% of previous stage	Notes	Possible reasons for losses relative to previous analysis stage
A	79564267	100		read-pairs from AYB base caller	
B	55047046	69.19	69.19	117 nt fragment reads recovered from combining 104 nt paired-end reads with 91 nt overlap as expected, with at most 6 mismatches within the overlap region	synthesis error, sequencing error, contamination
C	50145113	63.02	91.10	reads with indexing information indicating they belong to one of the five files encoded in this experiment	synthesis error or sequencing error leading to dinucleotide repeat, invalid file identification in indexing information or parity-check failure
of which:	18270252	22.96	33.19	file `watsoncrick.pdf`	
	8064484	10.14	14.65	file `wssnt10.txt`	
	11966357	15.04	21.74	file `EBI.jp2`	
	802908	1.01	1.46	file `View_huff3.cd.new`	
	11041112	13.88	20.06	file `MLK_excerpt_VBR_45-85.mp3`	
D	50141326	63.02	99.99	reads contributing 'votes' to final decoding of file	synthesis error or sequencing error in indexing information leading to invalid location in file
of which:	18269250	22.96	99.99	file `watsoncrick.pdf`	
	8063761	10.13	99.99	file `wssnt10.txt`	
	11965715	15.04	99.99	file `EBI.jp2`	
	802414	1.01	99.94	file `View_huff3.cd.new`	
	11040186	13.88	99.99	file `MLK_excerpt_VBR_45-85.mp3`	

Supplementary Table 3 | Numbers of reads used during decoding.



Supplementary Figure 3 | Distribution of sequencing depths over encoded locations. The mean is at a coverage of 1,308; the standard deviation is 459.

Majority voting was used to resolve any discrepancies caused by DNA synthesis or sequencing errors. The error rate amongst the 'votes' used to reconstruct the five files was 0.004004 (20.08M errors in 5,014M bases counted). This is higher than the combined synthesis and sequencing error rate reported above because of cases where one or a small number of errors in indexing information led to a read being misplaced in its correct file, or placed in an incorrect file, generating on average 75 incorrect votes (100 misplaced votes, each with probability ~0.75 of being incorrect).

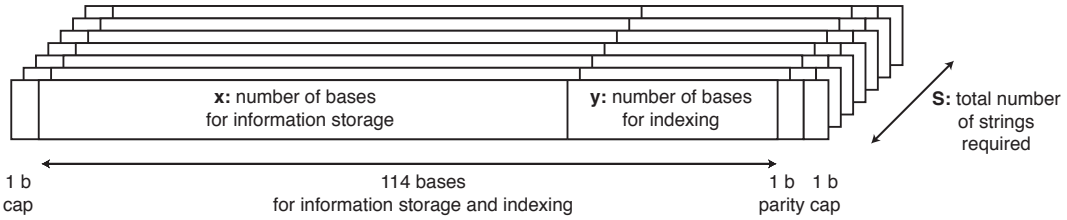
On completion of this procedure, four of the five original files were reconstructed perfectly. The fifth file required manual intervention to correct two regions each of 25 bases that were not recovered from any sequenced read, as described below.

Scaling properties of the DNA-storage scheme. In the following sections we demonstrate that, even constrained to today's technology, the scheme presented here scales nearly linearly, well beyond any realistically needed range, i.e. beyond 20 orders of magnitude larger than the estimated global amount of digital data of 3 ZB (3×10^{21} bytes).

The global data volume was estimated by adding the 1.8 ZB estimate of data produced in 2011 (ref. 22) to the estimated production in 2010, calculated assuming a doubling time of two years[22], to give 3 ZB ($= 1.8 + 1.8/\sqrt{2}$).

As each 117 base string can be synthesized and sequenced independently, it is reasonable to assume that the costs associated with these processes is linear in the number of strings. Therefore, to demonstrate the scaling behaviour of our scheme, we show that the number of strings required to reliably store the data increases nearly linearly with the amount of data. First, we focus on the relationship between the number of strings required and the amount of data to be stored. Second, we show that increase the amount of data to be stored does not lead to higher error rates. This is achieved by both theoretical and empirical estimates of the error rate as a function of amount of data and sequencing coverage.

Scaling of the total number of strings required. Let I be the information to be stored, in bytes. We now show that the number of 117 nt strings required to encode these data scales nearly linearly with respect to I. Recall that in each 117 nt string, 114 bases can be used to store data and indexing information. As the amount of data to be stored increases, we may need to use more than the current 14 bases for indexing. Supplementary Fig. 4 shows how, in general, the strings may be partitioned into x data bases and y indexing bases. Note that y depends on the total number of strings to be used, S. Optimally, $y = \lceil \log_3(S) \rceil$ and therefore $x = 114 - \lceil \log_3(S) \rceil$.



Supplementary Figure 4 | Schematic representation of information encoding in DNA. Multiple strings are used, each comprising 117 bases of which x may be used for storing information and y for indexing, with $x + y = 114$.

Intuitively, we can anticipate that the relationship between S and I is not linear: as the amount of information increases, the number of strings required also increases; this in turn requires more indexing information (and thus greater y), which leaves fewer bases to store information in each string (thus smaller x).

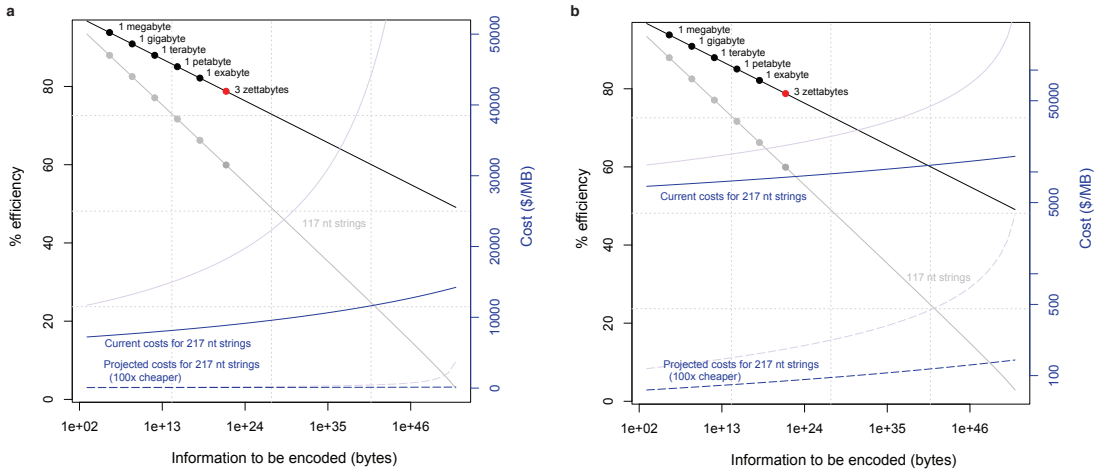
Expressed in bases, the information to be stored is $B = 5.07I$, as we can encode 5.07 bases/byte (see above). Since the encoding strings are 75% overlapping, the relationship between B and S is:

$$B = Sx / 4 = S(114 - \lceil \log_3(S) \rceil) / 4 \qquad (1)$$

This can be solved for S numerically. Fig. 2a (Main Text) illustrates the relationship of information stored (I) and the efficiency of encoding, measured as the proportion of bases synthesized (data and indexing) that are used to hold data ($eff = B / 117S$). Two features become apparent. First, the proportion of the total DNA available for encoding data decreases slowly, and is reasonable across the entire relevant data size range. Second, even constrained to 117 base-long strings, the current encoding scheme makes it possible to encode > 20 orders of magnitude more data than is currently practically relevant. (The theoretical limit comes when every base is required for indexing, and none remains to store information.)

DNA-storage costs are affected by the efficiency achieved for different information volumes. Current costs are about $12,400/MB (below), based on our efficiency of $B / 117S = 0.88$. Costs scale as the inverse of efficiency, and Fig. 2a (Main Text) also shows the cost function $12{,}400(0.88 / eff)$.

We repeated the above calculations based on longer synthesized strings. The Agilent OLS process can already produce 300-base oligos, with 244,000 designed strings costing approximately $30,000. Assuming that this would provide 217 nt strings for DNA-storage, an increase of 100 nt, we get twice as many available bases for 30/25 times the price. This would give a current cost of about $7,440/MB (= 12,400 × (30,000 / 25,000) / 2) based on an efficiency of 0.94 (the encoding efficiency that would be achieved repeating our experiment with 217 nt strings) and consequently a cost function of $7{,}440(0.94 / eff)$. Supplementary Fig. S5 repeats the information of Fig. 2a (Main Text), and adds the corresponding results for these longer strings. This shows that with achievable improvement in DNA synthesis technology the scalability of DNA-storage is substantially improved, with higher efficiency, lower cost and slower decline in efficiency and increase in cost for larger data volumes.



Supplementary Figure 5 | Scaling properties of DNA-storage. The graphs show how encoding efficiency and costs change as the amount of stored information increases, for longer strings with 217 nt available for data and indexing. The *x*-axis (logarithmic scale) represents the total amount of information to be encoded. Common data scales are indicated, including the 3 ZB global data estimate. The black line (*y*-axis scale to left) indicates encoding efficiency, measured as the proportion of synthesized bases available for data encoding. The blue curves (*y*-axis scale to right) indicate the corresponding effect on encoding costs, both at current synthesis cost levels (solid line) and in the case of a two-order of magnitude reduction (dashed line). The pale grey and pale blue lines give the corresponding results for 117 nt strings, for ease of comparison with Fig. 2a (Main Text). **a**, Linear cost scale; **b**, logarithmic cost scale.

Scaling of the decoded data error rate. Assuming that all synthesized strings have the same probability of being sequenced, the mean error rate per base of encoded data depends on three variables:

- ε: the mean error rate per base at the level of a sequencing read (due to synthesis and sequencing error), set to 0.004 as determined for our experiment (above)
- S: the total number of designed strings
- c_B: the base coverage (mean number of times each base of encoded information is sequenced)

Recall that the decoding scheme calls each base of encoded information based on a majority vote of all the read bases corresponding to its position. Because each base of encoded information is represented in four strings (due to the 75% overlap in encoded data between neighbouring strings), the mean string coverage is $c_S = c_B / 4$. Thus, in total, there are $Sc_S = Sc_B / 4$ reads. The probability that any read covers base i of encoded information is $4 / S$. Thus, the number x_i of base reads for encoded base i follows a binomial distribution with number of trials $Sc_B / 4$ and probability of success $4 / S$, which we write as $B(Sc_B / 4, 4 / S)$.

Next, consider that for encoded base i to be correctly called, the majority of the x_i read bases (votes) need to be correct. The distribution of correct bases $x_{i,\text{correct}}$ is $B(x_i, 1 - \varepsilon)$. The majority vote regarding encoded base i is wrong if $x_{i,\text{correct}} < x_i / 2$. (This is the worst case scenario that all incorrect votes are for the same incorrect base. Our results are not significantly altered when other, more-favourable, scenarios are considered.) Because of this

dependency, the expected encoded base error rate is not straightforward to compute analytically, but it is easily estimated by Monte Carlo simulation. Supplementary Fig. 6 shows an R function that performs this estimation.

```
# R function to estimate the encoded base error rate:
estimateBaseErrorRateMC = function(eps,S,cS,nsamples) {
  x = rbinom(nsamples, cS*S ,4/S);
  s = sapply(x,function(t) rbinom(1,t,1-eps));
  e = 1 - sum (s > x/2) / nsamples;
  return(e);
}
```

Supplementary Figure 6 | R function to estimate encoded base error rate.

Supplementary Table 4 provides estimates of the encoded base error rate, as a function of data size and sequencing effort (per encoded byte) relative to our experiment, based on 10^5 Monte Carlo samples (`nsamples` in Supplementary Fig. 6) per cell. This shows that, keeping sequencing effort per encoded byte constant, the error rate increases only very slowly with the increase in amount of data.

Error rate as function of data size and sequencing effort

% sequencing effort (per encoded byte) relative to experiment	Information size in byte																	
	1000	1MB	1GB	1TB	1PB	1EB	1ZB	1.00E+24	1.00E+27	1.00E+30	1.00E+33	1.00E+36	1.00E+39	1.00E+42	1.00E+45	1.00E+48
1000%	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
316%	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100%	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31%	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10%	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3%	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00E-05	0.00038	0.00479
1%	0	0	1.00E-05	4.00E-05	3.00E-05	7.00E-05	9.00E-05	0.00016	0.00054	0.00118	0.00317	0.00657	0.0145	0.03268	0.07506	0.17551
0.3%	0.01221	0.01764	0.02209	0.02725	0.03568	0.04084	0.05129	0.06733	0.08702	0.11427	0.14752	0.19205	0.24854	0.32076	0.42418	0.5758
0.1%	0.24756	0.2693	0.29461	0.31652	0.34522	0.34376	0.37356	0.40983	0.44767	0.50255	0.54485	0.59052	0.63929	0.69121	0.7514	0.81493
0.03%	0.63294	0.66131	0.67665	0.69713	0.71112	0.70758	0.72623	0.74469	0.76418	0.78336	0.80056	0.8224	0.84375	0.86439	0.88685	0.90861
0.01%	0.859	0.87701	0.88289	0.89178	0.90067	0.87071	0.8787	0.88407	0.89109	0.89681	0.90474	0.91015	0.91856	0.9251	0.93259	0.94003
0.003%	0.9575	0.95954	0.96263	0.96321	0.96711	0.92772	0.9297	0.9313	0.93422	0.9357	0.93776	0.94094	0.9441	0.94557	0.94769	0.94916
0.001%	0.97811	0.98709	0.98762	0.9881	0.98941	0.99047	0.94677	0.94583	0.9464	0.94967	0.94959	0.94987	0.95178	0.95045	0.95207	0.95358

Supplementary Table 4 | Error rate as a function of data size and sequencing effort. Percentage sequencing effort is measured relative to our experiment; the highlighted row corresponds to the same sequencing effort per encoded byte as realised in our experiment.

In contrast, the error rate depends strongly on the coverage. Supplementary Table 4 suggests that the effective coverage of our actual experiment (1,308×; see above) could be substantially lowered without impacting on the error rate. To confirm this theoretical analysis using our empirical data, we subsampled the 79.6M read-pairs at varying fractions and attempted to reconstruct the five encoded files using our original protocol. Fig. 2b (Main Text) presents results on the per-encoded-base error rate for recovery of the file `watsoncrick.pdf`, which always has at least 50 base errors due to encoded bases that were not recovered from any sequenced read (see below), for the recovery of the other four encoded files combined and for our theoretical predictions based on the analysis above. The plot shows the error rate (y-axis, as a percentage) as a function of subsampling percentage (x-axis, logarithmic scale). This indicates good agreement of our theoretical and empirical

results. The difference between the `watsoncrick.pdf` results and the other four files is explained by the unrecoverable 50 bases described above. In this case, the minimum possible error rate is 0.0036% (10 bytes not recovered, out of 280,864). The discrepancy between the theoretical and empirical curves for relatively high subsampling fractions is probably due to model violation. For example, unlike in our model, the true sampling probability of strings is almost certainly not uniform due to unequal DNA synthesis and sequencing efficiencies. Note however that the discrepancy corresponds to a difference of only a few per cent in terms of reads used, and the model remains a good approximation. The plots confirm that we could reduce sequencing coverage by a factor of 10 or even 100 without significantly impacting on our ability to recover the encoded information.

Modelling cost-effectiveness of DNA-storage. We modelled the costs of DNA-storage over time according to:

$$C_D(t) = D_0 + Ft \tag{2}$$

where $C_D(t)$ is the cost to archive 1 MB of information for a period of t years. D_0 is the initial cost to write this information to DNA-storage — this will decrease over time as DNA synthesis technology improves — and F is the cost per year to maintain a DNA-storage facility (per MB of information stored). Storage on magnetic tape was modelled according to:

$$C_T(t) = T_0 + Ft + \sum_{i=1}^{ft}\left[R_{fix} + \frac{T_0 + R_{dim}}{\left(2^{f^{-1}/2.5}\right)^i}\right] \tag{3}$$

where $C_T(t)$ is the cost to archive 1 MB. T_0 is the initial cost to write this information to tape and F is the cost per year to maintain a tape storage facility (e.g. data centre), assumed equal to the corresponding cost for a DNA-storage archive. This assumption is likely to strongly favour tape over DNA due to the costs of power and of recurrent replacement of computing and tape hardware. The parameter f is the frequency of 'tape transfer events', i.e. how often it is necessary to read and re-write the information using the current technology as the previous one becomes obsolete. Industry standards suggest f is likely to be approximately $^1/_5$–$^1/_{10}$ yr^{-1}, i.e. data must be read and re-written to new technology every 5–10 years[49,50]. The summation represents the costs (per MB) of the ft transfer events occurring in t years, each comprising fixed cost R_{fix} (e.g. finite labour cost of retrieval of existing tape archive, set-up of copying process, storage of new archive material) and diminishing costs proportional to T_0 (for new storage media) and R_{dim} (other expenses, e.g. costs proportional to time spent reading and re-writing information), both of which are assumed to halve every 2.5 years due to technological improvements[51].

Tape costs are already very low, and so we set $T_0 = 0$. The break-even point when DNA-storage achieves the same cost as tape comes when $C_D(t) = C_T(t)$; equating equations (2) and (3) allows us to write:

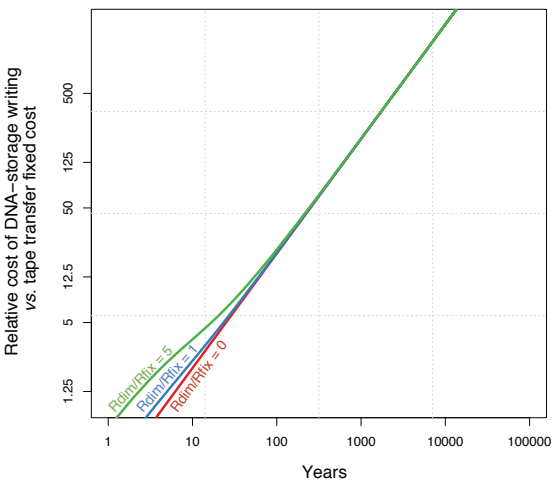
$$\frac{D_0}{R_{fix}} = ft + \frac{R_{dim}}{R_{fix}}\left(\frac{1 - 2^{-t/2.5}}{2^{f^{-1}/2.5} - 1}\right) \tag{4}$$

D_0/R_{fix} is the relative cost of writing DNA-storage compared to the fixed costs of tape transfer events. Equation (4) indicates the balance between values of D_0, f, t, R_{fix} and R_{dim} that leads to

break-even point of DNA-storage; smaller values of D_0/R_{fix} or greater values of t correspond to conditions where DNA-storage is more cost-effective than tape; conversely, larger D_0/R_{fix} or smaller t make tape favourable.

The current commercial cost of the Agilent OLS process is approximately $25,000 for 244,000 designed oligos of length 200 bases (approximately $0.05/100 bases). We encoded 739 kB in 153,335 DNA strings of length 117 bases, leading to a value for D_0 of approximately $12,400/MB (= 25,000/(0.739 × (244,000/153,335) × (200/117))). For archives of a few megabytes, we estimate that the cost in personnel, labour and management of a corresponding tape technology transition might be of the order of $25–100, leading to a current estimate of D_0/R_{fix} in the range 125–500. Other current DNA synthesizing methods, e.g. maskless photolithography, can be used to produce shorter oligos with potentially higher error rates[52], but are less expensive per base synthesized. It is possible that these could be used to reduce the cost per MB (D_0).

For realistic values of f (above), the second term on the right-hand side of equation (4) rapidly becomes small. Supplementary Fig. 7 shows the relationship of D_0/R_{fix} and break-even points (timescale on which DNA-storage and tape storage costs equate) when R_{dim}/R_{fix} = 0, 1 and 5 and $f = {}^1/_5$. It is clear that even for R_{dim}/R_{fix} = 5, which is unrealistically large, the effect of R_{dim}/R_{fix} is negligible for values of D_0/R_{fix} that are likely to be achieved in the near future (see Main Text). The same is true for $f = {}^1/_{10}$ (not shown). Consequently, we have assumed R_{dim}/R_{fix} = 1 for illustrative purposes. Fig. 2c (Main Text) plots D_0/R_{fix} against time t in this case, highlighting the break-even points for $f = {}^1/_5$ and $f = {}^1/_{10}$.



Supplementary Figure 7 | Effect of R_{dim}/R_{fix} on DNA-storage break-even timescale. The x-axis is the break-even time beyond which DNA-storage is less expensive than magnetic tape, assuming the tape archive has to be read and re-written every 5 years ($f = {}^1/_5$); the y-axis is the relative cost of DNA-storage synthesis and tape transfer fixed costs. Lines plotted are for R_{dim}/R_{fix} = 0 (red), 1 (blue) and 5 (green). Note the logarithmic scales on both axes.

Information decoding costs. In our experiment, we decoded 739 kB of information using one lane of the Illumina HiSeq 2000, at a sequencing cost of approximately $1,600. This gives a decoding cost of ~$2,200/MB (= 1,600/0.739). As shown above, we could have sequenced 10 times as much encoded information in the same run and still recovered our data. This suggests that ~$220/MB (= 2,200/10) is a reasonable approximation for the decoding costs for optimised use of existing technologies.

SUPPLEMENTARY DISCUSSION

Repair of file with missing reads. During decoding, one file (ultimately determined to be `watsoncrick.pdf`) reconstructed *in silico* at the level of DNA (prior to decoding, via base-3, to bytes) contained two regions, each of 25 bases in length, that were not recovered from any sequenced read. Given the overlapping segment structure of our encoding, each such region indicates the failure of any oligo representing any of four consecutive segments to be synthesized or sequenced successfully, as any one of four consecutive overlapping segments would have contained the bases corresponding to this location. Inspection of the two regions indicated that the non-detected bases fell within long repeats of the following 20-base motif:

```
5′-GAGCATCTGCAGATGCTCAT-3′
```

(colours used to highlight motif repeats; see below). We noticed that repeats of this motif have a self-reverse complementary pattern (Supplementary Fig. 8) and we hypothesised that long, self-reverse complementary DNA fragments might not be readily sequenced using the Illumina process. In terms of DNA synthesis, we know no reason to expect self-reverse complementary fragments to be problematic in the Agilent OLS system. The PCR conditions used for library construction involved denaturing the template oligos at 98 °C, initially for a period of 3 min and for 80 s per cycle thereafter. These conditions are more than sufficient to denature and amplify any self-annealing oligos. A further denaturing step separates the double-stranded products prior to dilution and flowcell loading, and bridge amplification should also proceed nominally to form clusters of clonally amplified library constructs.

```
  5′-...GAGCATCTGCAGATGCTCATGAGCATCTGCAGATGCTCATGAGCATCTGCAGATGCTCAT...-3′
        ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
3′-...TACTCGTAGACGTCTACGAGTACTCGTAGACGTCTACGAGTACTCGTAGACGTCTACGAG...-5′
```

Supplementary Figure 8 | Self-reverse complementary nature of the 20-base motif.

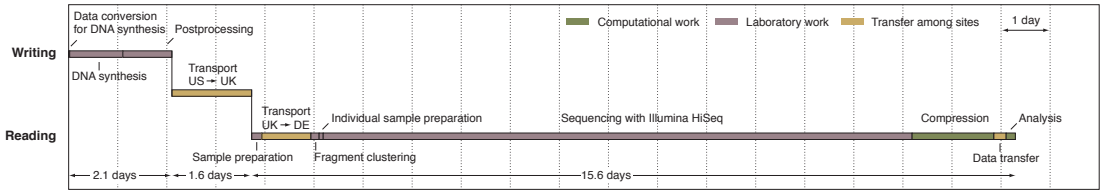
However, the subsequent sequencing conditions are intended to promote DNA hybridisation/annealing, and so in principle internal secondary structures might form and remain stable throughout the process. We therefore reason that each of the two self-complementary regions produced by our encoding scheme led to the formation of stem-loop structures within the target sequences, inhibiting the sequencing-by-synthesis reaction over these nonlinear stretches. The associated clusters would be omitted from the imaging readout, thereby resulting in a gap in the sequencing data when the files were reconstructed.

Examination of our coding methods indicated that such long repeats of the 20-base motif would arise when the original computer file contained repeats of byte value 255 (hexadecimal FF). Consequently, we modified the *in silico* reconstructed DNA sequence to repair the repeating motif pattern and subjected this to subsequent decoding steps. No further problems were encountered, and the final decoded file matched perfectly the file `watsoncrick.pdf`.

With hindsight, we should have devised a code that ensured that no long self-complementary regions existed in any of our designed DNA segments. One way to achieve this would be to pre-process the files to be encoded using a one-time pad or other stream

cipher with a standard or known key stream, leading to the DNA segments having random properties[53].

Timescale of DNA-storage experiment. Supplementary Fig. 9 shows the time taken for each stage of our DNA-storage experiment. The experiment was not optimised for speed. All encoding and decoding computations were performed on one core of an Intel i5-2540M processor running at 2.60 GHz, except for the reconstruction of full-length (117 base) oligos from paired-end (104 base) reads which was performed using one core of an Intel Xeon X5650 at 2.67GHz. In a large-scale DNA-storage archive, transfer periods could be eliminated by having encoding, storage and decoding taking place at one site. Both computer software and laboratory procedures could readily be optimised and parallelised; laboratory procedures could also be automated with liquid-handling robotics for high-throughput applications[54]. Both computational and laboratory equipment are subject to continual innovation, improving their speed.



Supplementary Figure 9 | Timeline of DNA-storage experiment. We report only periods of active work on the experiment. We have omitted time taken to devise repairs for the file with two information gaps (above).

Information storage density. We recovered 757,051 bytes of information from 337 pg of DNA (above), giving an information storage density of ~2.2 PB/g (= 757,051/337 × 10^{-12}). We note that this information density is enough to store the US National Archives and Records Administration's Electronic Records Archives' 2011 total of ~100 TB (ref. 55) in < 0.05 g of DNA, the Internet Archive Wayback Machines's 2 PB archive of web sites[56] in ~1 g of DNA, and CERN's 80 PB CASTOR system for LHC data[25] in ~35 g of DNA.

REFERENCES FOR SUPPLEMENTARY INFORMATION

29. Leier, A., Richter, C., Banzhaf, W. & Rauhe, R. Cryptography with DNA binary strands. *Biosystems* **57**, 13–22 (2000)
30. Bancroft, C., Bowler, T., Bloom, B. & Clelland, C. T. Long-term storage of information in DNA. *Science* **293**, 1763–1765 (2001)
31. Wong, P. C., Wong, K.-K. & Foote, H. Organic data memory. Using the DNA approach. *Comm. ACM* **46**, 95–98 (2003)
32. Arita, M. & Ohashi, Y. Secret signatures inside genomic DNA. *Biotechnol. Prog.* **20**, 1605–1607 (2004)
33. Kashiwamura, S., Yamamoto, M., Kameda, A., Shiba, T. & Ohuchi, A. Potential for enlarging DNA memory: the validity of experimental operations of scaled-up nested primer molecular memory. *Biosystems* **80**, 99–112 (2005)

34. Skinner, G. M., Visscher, K. & Mansuripur, M. Biocompatible writing of data into DNA. *J. Bionanoscience* **1**, 17–21 (2007)
35. Yachie, N., Sekiyama, K., Sugahara, J., Ohashi, Y. & Tomita, M. Alignment-based approach for durable data storage into living organisms. *Biotechnol. Prog.* **23**, 501–505 (2007)
36. Heider, D. & Barnekow, A. DNA watermarks: a proof of concept. *BMC Mol. Biol.* **9**, 40 (2008)
37. Portney, N. G., Wu, Y., Quezada, L. K., Lonardi, S. & Ozkan, M. Length-based encoding of binary data in DNA. *Langmuir* **24**, 1613–1616 (2008)
38. CUHK iGEM 2010. Bacterial-based storage and encryption device (2010) http://2010.igem.org/Team:Hong_Kong-CUHK accessed online, 10 May 2012
39. Jarvis, C. & Netherlands Proteomics Centre. Blighted by Kenning (2012) http://www.artforeating.co.uk/restaurant/index.php?/blighted-by-ken/project-overview/ accessed online, 8 November 2012
40. Yamamoto, M., Kashiwamura, S., Ohuchi, A. & Furukawa, M. Large-scale DNA memory based on the nested PCR. *Natural Computing* **7**, 335–346 (2008)
41. Kosuri, S. *et al.* A scalable gene synthesis platform using high-fidelity DNA microchips. *Nature Biotech.* **28**, 1295–1299 (2010)
42. Beaucage, S. L. & Caruthers, M. H. Deoxynucleoside phosphoramidites — a new class of key intermediates for deoxypolynucleotide synthesis. *Tetrahedron Lett.* **22**, 1859–1862 (1981)
43. Cleary, M. A. *et al.* Production of complex nucleic acid libraries using highly parallel *in situ* oligonucleotide synthesis. *Nature Methods* **1**, 241–248 (2004)
44. Aird, D. *et al.* Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome Biol.* **12**, R18 (2011)
45. Andrews, S. FastQC. A quality control tool for high throughput sequence data (2012) http://www.bioinformatics.babraham.ac.uk/projects/fastqc accessed online, 5 September 2012
46. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* **25**, 1754–1760 (2009)
47. McKenna, A. *et al*. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res*. **20**, 1297–1303 (2010)
48. Minoche, A. E., Dohm, J. C. & Himmelbauer, H. Evaluation of genomic high-throughput sequencing data generated on Illumina HiSeq and Genome Analyzer systems. *Genome Biol*. **12**, R112 (2011)
49. Vican, I. & Stančić, H. Long-term inactive data retention through tape storage technology. *In* Stančić, H. *et al.* (eds.) *INFuture2009: the Future of Information Sciences — Digital Resources and Knowledge Sharing*. pp. 105–114. (Department of Information Sciences, University of Zagreb, 2009)
50. Klingler, S. L. Information storage technologies. *In* Bates, M. J. (ed.) *Understanding Information Retrieval Systems: Management, Types and Standards.* pp. 245–258. (CRC Press, 2012)
51. Fujifilm. Fujifilm to manufacture 5TB tape cartridge for Oracle's StorageTek T10000C drive (2011) http://www.fujifilm.com/news/n110201.html accessed online, 10 May 2012
52. Agbavwe, C. *et al.* Efficiency, error and yield in light-directed maskless synthesis of DNA microarrays. *J. Nanobiotechnology* **9**, 57 (2011)
53. Paar, C. & Pelzl, J. *Understanding Cryptography. A Textbook for Students and Practitioners.* (Springer, 2010)

54. Quail, M. A., Swerdlow, H. & Turner, D. J. Improved protocols for the Illumina genome analyzer sequencing system. *Curr. Protoc. Hum. Genet.* **62**, 18.2.1–18.2.27 (2009)
55. US National Archives & Records Administration (2012) http://www.archives.gov/era/status.html accessed online, 10 May 2012
56. Internet Archive. The Wayback Machine (2012) http://archive.org/about/faqs.php accessed online, 7 September 2012

SUPPLEMENTARY INFORMATION
doi:10.1038/nature11875

Method for Encoding and Decoding Arbitrary Computer Files in DNA Fragments

1 Encoding

1.1: An arbitrary computer file is represented as a string $S_\emptyset$ of bytes (often interpreted as a number between $\emptyset$ and $2^8 - 1$, i.e. a value in the set $\{\emptyset \ldots 255\}$)*.

1.2: $S_\emptyset$ is encoded using a given Huffman code, converting it to base-3. This generates the string S_1 of characters in $\{\emptyset, 1, 2\}$, each such character called a 'trit'.

1.3: Write $len()$ for the function that computes the length (in characters) of a string, and define $n = len(S_1)$. Represent n in base-3 and prepend $\emptyset$s to generate a string S_2 of trits such that $len(S_2) = 2\emptyset$. Form the string concatenation $S_4 = S_1 \,.\, S_3 \,.\, S_2$, where S_3 is a string of at most 24 $\emptyset$s chosen so that $len(S_4)$ is an integer multiple of 25.

1.4: S_4 is converted to a DNA string S_5 of characters in $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ with no repeated nucleotides (nt) using the scheme illustrated in Figure 1. (The first trit of S_4 is coded using the 'A' row of the table. For each subsequent trit, characters are taken from the row defined by the previous character conversion.)

previous nt written	next trit to encode		
	∅	1	2
A	C	G	T
C	G	T	A
G	T	A	C
T	A	C	G

Figure 1: Base-3 to DNA encoding ensuring no repeated nucleotides. For each trit t to be encoded, select the row labelled with the previous nt used and the column labelled t and encode using the nt in the corresponding table cell.

1.5: Define $N = len(S_5)$, and let ID be a 2-trit string identifying the original file and unique within a given experiment (permitting mixing of DNA from different

*$\emptyset$ is used throughout to represent the number zero, to avoid confusion with letters o and O.

files $S_{\emptyset}$ in one experiment). Split S_5 into overlapping segments of length 1ØØ nt, each offset from the previous by 25 nt. This means there will be $\frac{N}{25} - 3$ segments, conveniently indexed $i = \emptyset \ldots \frac{N}{25} - 4$; segment i is denoted F_i and contains (DNA) characters $25i \ldots 25i + 99$ of S_5.

Each segment F_i is further processed as follows:

1.6: If i is odd, reverse complement F_i.

1.7: Let $i3$ be the base-3 representation of i, appending enough leading Øs so that $len(i3) = 12$. Compute P as the sum (mod 3) of the odd-positioned trits in ID and $i3$, i.e. $ID_1 + i3_1 + i3_3 + i3_5 + i3_7 + i3_9 + i3_{11}$. ($P$ acts as a 'parity trit'—analogous to a parity bit—to check for errors in the encoded information about ID and i.)

1.8: Form the indexing information string $IX = ID \,.\, i3 \,.\, P$ (comprising 2+12+1 = 15 trits). Append the DNA-encoded version of IX to F_i using the same strategy as at step (1.4) above, starting with the code table row defined by the last character of F_i, to give indexed segment F_i'.

1.9: Form F_i'' by prepending A or T and appending C or G to F_i'—choosing between A and T, and between C and G, randomly if possible but always such that there are no repeated nt. (This ensures that we can distinguish a DNA segment that has been reverse complemented during DNA sequencing from one that has not—the former will start with G|C and end with T|A; the latter will start A|T and end C|G.)

See Figure 2 for a schematic representation of the DNA-encoding of computer files.

1.10: The segments F_i'' are synthesized as actual DNA oligonucleotides and stored, and may be supplied for sequencing.

2 Example

2.1: As it is difficult to represent all possible bytes in this document, we use a simple example of a file comprising just 18 bytes that happen to be easily

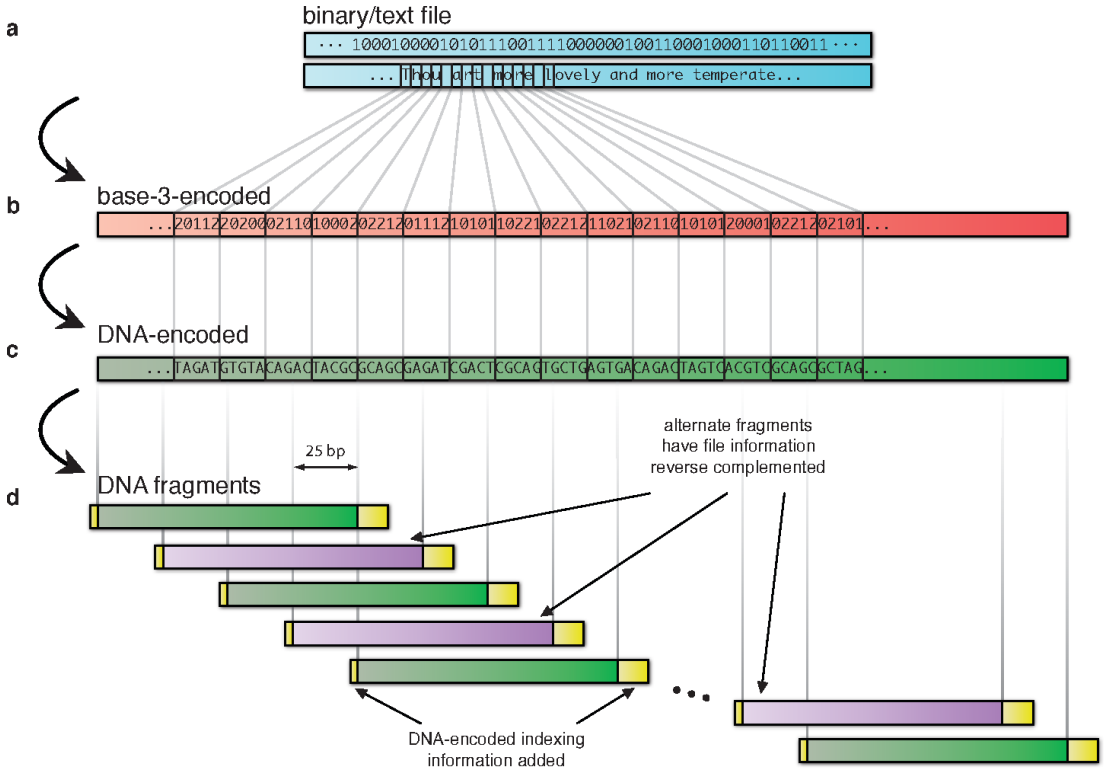


Figure 2: Schematic of DNA-fragment-encoding of computer files. The computer file (in any format, e.g. binary or text) shown in blue in (a) (step 1.1 above) is encoded in base-3 (red in b; 1.2–1.3) and then as DNA with no repeated nt (green in c; 1.4). This representation of the complete file is then split into overlapping segments (1.5), alternate segments reverse complemented (mauve; 1.6), and segment indexing and direction-determining information added (yellow; see d; 1.7–1.9).

represented via ASCII codes (for clarity, we show spaces as ␣):

$$S_{\emptyset} = \texttt{Birney␣and␣Goldman}$$

2.2: Using the Huffman code defined in the example file `View_huff3.cd`, we convert the bytes of $S_{\emptyset}$ (shown above/below S_1 for illustrative purposes) into base-3[†]:

[†]Throughout what follows, spaces are not part of base-3 or DNA strings but are included to assist in 'reading' how strings have been derived in each step.

	B	i	r	n	e	y	␣	a	n
S_1 =	2Ø1ØØ	2Ø21Ø	1Ø1Ø1	ØØØ21	2ØØØ1	222111	Ø2212	Ø1112	ØØØ21
	221ØØ	Ø2212	222212	Ø211Ø	Ø21Ø1	221ØØ	11Ø21	Ø1112	ØØØ21
	d	␣	G	o	l	d	m	a	n

2.3: $n = len(S_1) = 92$, which is 1Ø1Ø2 in base-3. So:

$$S_2 = \text{ØØØØØØØØØØØØØØØ}\ \ \text{1Ø1Ø2 (length 2Ø)}$$

$$S_3 = \text{ØØØØØØØØØØØØØ (length 13)}$$

$$\begin{aligned} S_4 &= S_1\ .\ S_3\ .\ S_2 \\ &= \text{2Ø1ØØ2Ø21Ø1Ø1Ø1ØØØ212ØØØ1222111Ø2212Ø1112ØØØ21} \\ &\quad \text{221ØØØ221222221Ø211ØØ21Ø1221ØØ11Ø21Ø1112ØØØ21} \\ &\quad \text{ØØØØØØØØØØØØØ ØØØØØØØØØØØØØØØ1Ø1Ø2} \\ &\quad \text{(length } 92 + 13 + 2Ø = 125 = 5 \times 25\text{)} \end{aligned}$$

2.4: Using the DNA coding strategy and table shown at step (1.4) above, convert S_4 to DNA:

$$\begin{aligned} S_5 &= \text{TAGTATATCGACTAGTACAGCGTAGCATCTCGCAGCGAGATACGCTGCTACGCAGCATGCTGT} \\ &\quad \text{GAGTATCGATGACGAGTGACTCTGTACAGTACGTACGTACGTACGTACGTACGACTAT} \end{aligned}$$

2.5: $N = len(S_5) = 125$, and we choose (e.g.) $ID = 12$. S_5 will be split into overlapping segments F_i of length 1ØØ nt for $i \in \{Ø \ldots \frac{125}{25} - 4\}$, i.e. $i \in \{Ø, 1\}$. With overlapping parts underlined for illustration, $F_Ø$ and F_1 are:

$$\begin{aligned} F_Ø &= \text{TAGTATATCGACTAGTACAGCGTAG } \underline{\text{CATCTCGCAGCGAGATACGCTGCTA}} \\ &\quad \underline{\text{CGCAGCATGCTGTGAGTATCGATGA}}\ \underline{\text{CGAGTGACTCTGTACAGTACGTACG}} \end{aligned}$$

F_1 = CATCTCGCAGCGAGATACGCTGCTA CGCAGCATGCTGTGAGTATCGATGA

CGAGTGACTCTGTACAGTACGTACG TACGTACGTACGTACGTACGACTAT

2.6: Only $i = 1$ is odd, so F_1 is reverse complemented:

F_1 = ATAGTCGTACGTACGTACGTACGTACGTACGTACTGTACAGAGTCACTCG

TCATCGATACTCACAGCATGCTGCGTAGCAGCGTATCTCGCTGCGAGATG

2.7: For $i = \emptyset$, $i3 = \emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset$ (length 12) and the sum (mod 3) of the odd-positioned trits of ID and $i3$ is $P = 1 + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset \pmod 3 = 1$. For $i = 1$, $i3 = \emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset 1$ and $P = 1 + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset + \emptyset \pmod 3 = 1$.

2.8: For $i = \emptyset$, $IX = ID\ .\ i3\ .\ P = 12\ \emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\ 1$; for $i = 1$, $IX = 12\ \emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset\emptyset 1\ 1$. So:

$F'_\emptyset$ = TAGTATATCGACTAGTACAGCGTAGCATCTCGCAGCGAGATACGCTGCTA

CGCAGCATGCTGTGAGTATCGATGACGAGTGACTCTGTACAGTACGTACG

AT ACGTACGTACGT C (length $1\emptyset\emptyset + 15 = 115$)

F'_1 = ATAGTCGTACGTACGTACGTACGTACGTACGTACTGTACAGAGTCACTCG

TCATCGATACTCACAGCATGCTGCGTAGCAGCGTATCTCGCTGCGAGATG

AT ACGTACGTACGA G (length 115)

2.9: Prepend A|T and append C|G (note that in this example we have only one random choice, at the end of $F''_\emptyset$):

$F''_\emptyset$ = A TAGTATATCGACTAGTACAGCGTAGCATCTCGCAGCGAGATACGCTGCTA

CGCAGCATGCTGTGAGTATCGATGACGAGTGACTCTGTACAGTACGTACG

ATACGTACGTACGTC G (length $1 + 115 + 1 = 117$)

F''_1 = T ATAGTCGTACGTACGTACGTACGTACGTACGTACTGTACAGAGTCACTCG

TCATCGATACTCACAGCATGCTGCGTAGCAGCGTATCTCGCTGCGAGATG

ATACGTACGTACGAG C (length 117)

3 Decoding

Decoding is simply the reverse of encoding, starting with sequenced DNA fragments F_i'' of length 117 nt. Reverse complementation during the DNA sequencing procedure (e.g. during PCR reactions) can be identified for subsequent reversal by observing whether fragments start with A|T and end with C|G, or start G|C and end T|A. With these two 'orientation' nt removed, the remaining 115 nt of each segment can be split into the first 1ØØ 'message' nt and the remaining 15 'indexing' nt. The indexing nt can be decoded to determine the file identifier *ID* and the position index *i3* and hence i, and errors may be detected by testing the parity trit P. Position indexing information permits the reconstruction of the DNA-encoded file, which can then be converted to base-3 using the reverse of the encoding table in step (1.4) above and then to the original bytes using the given Huffman code. Precise details of the decoding procedure are left as an exercise for the reader.

Errors introduced during DNA synthesis, storage or sequencing could lead to various artefacts, particularly nt insertion, deletion or substitution. Recovery of information from fragments with such errors may be possible (details left as exercise)—we have not found this to be necessary due to the large numbers of perfectly-sequenced fragments available via high-throughput sequencing.

doi:10.1038/nature11875

FastQC Report

Tue 4 Sep 2012
aln.bam

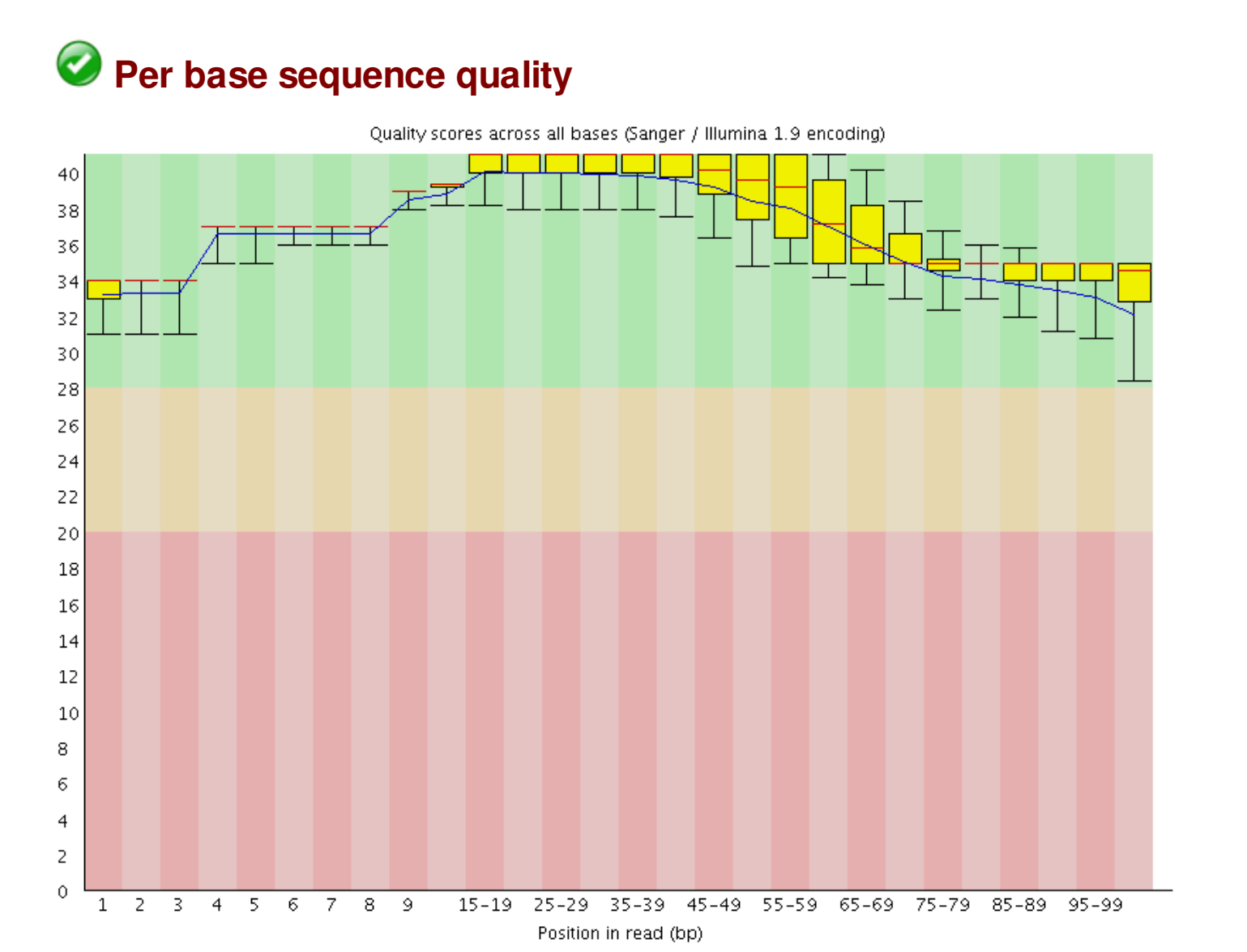
Summary

- Basic Statistics
- Per base sequence quality
- Per sequence quality scores
- Per base sequence content
- Per base GC content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Kmer Content

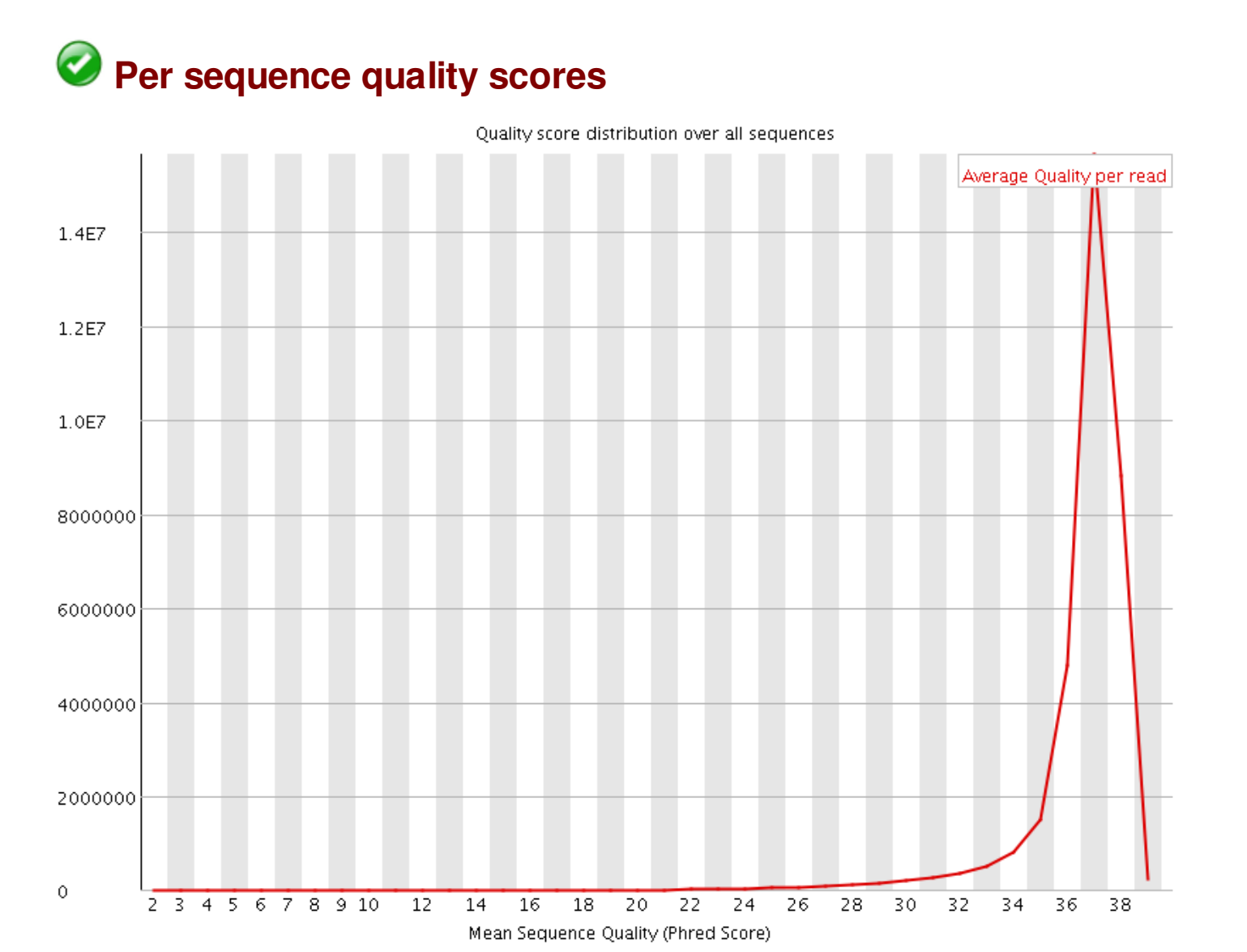
Basic Statistics

Measure	Value
Filename	aln.bam
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	34252010
Filtered Sequences	0
Sequence length	104
%GC	49

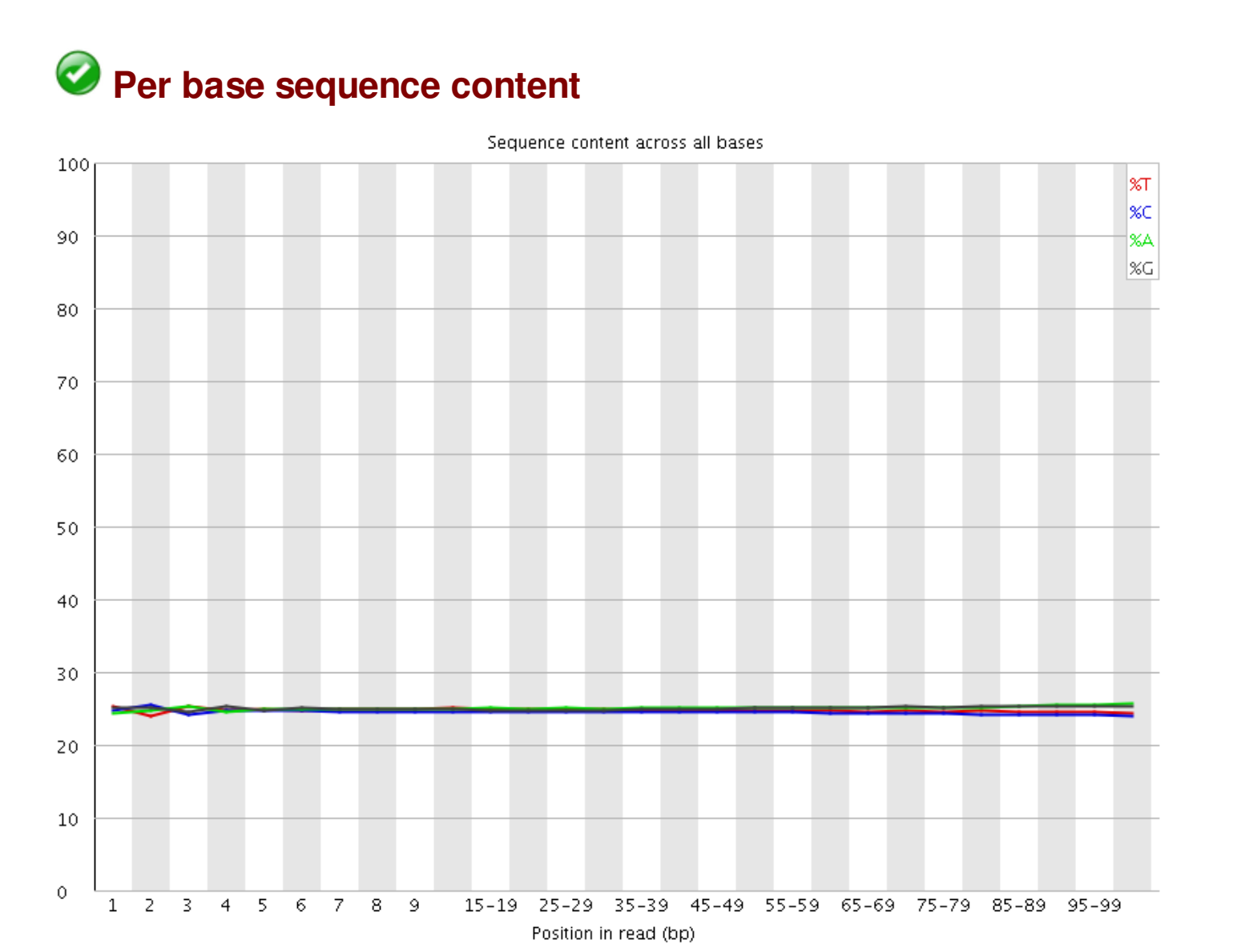
Per base sequence quality



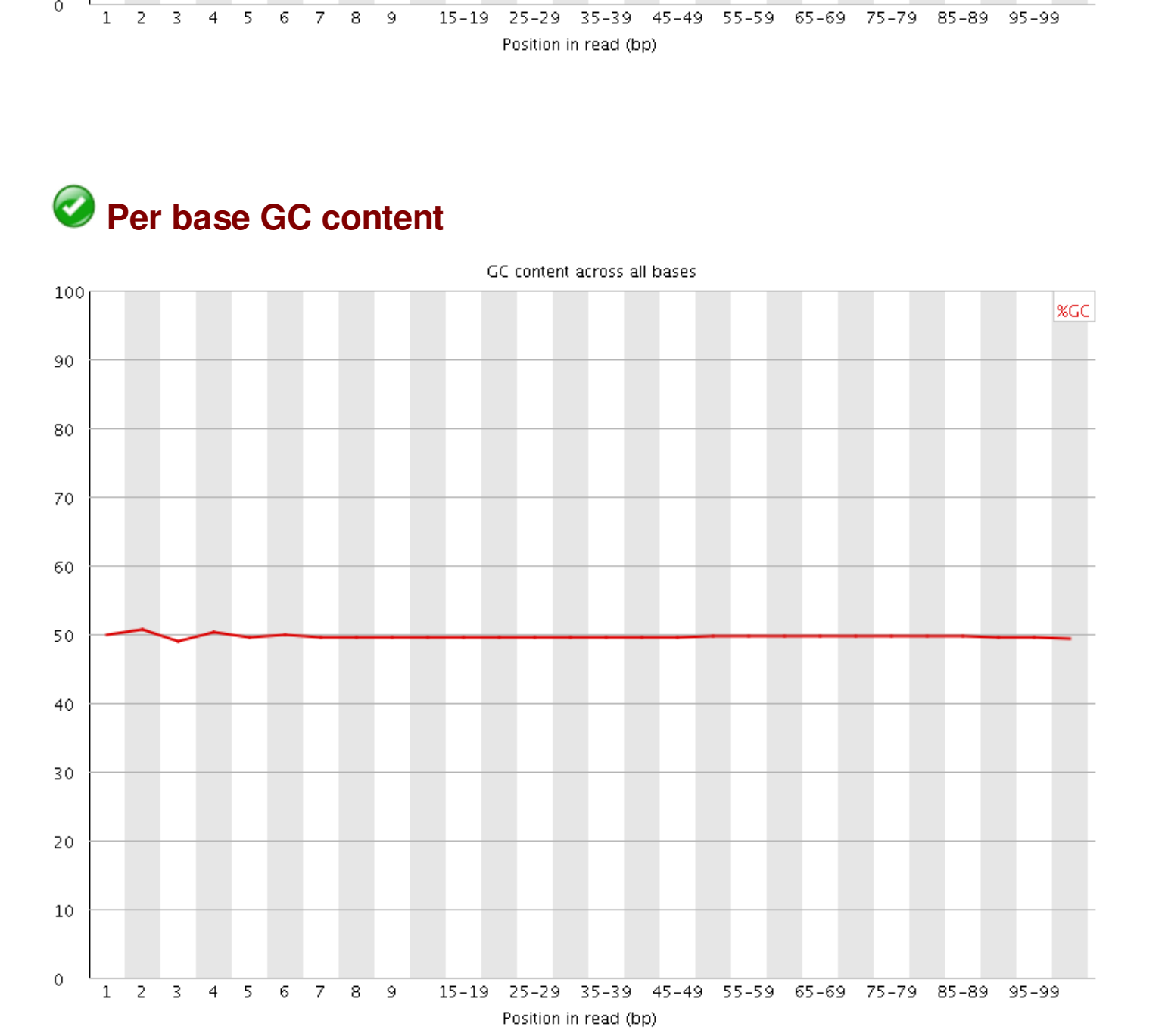
Per sequence quality scores



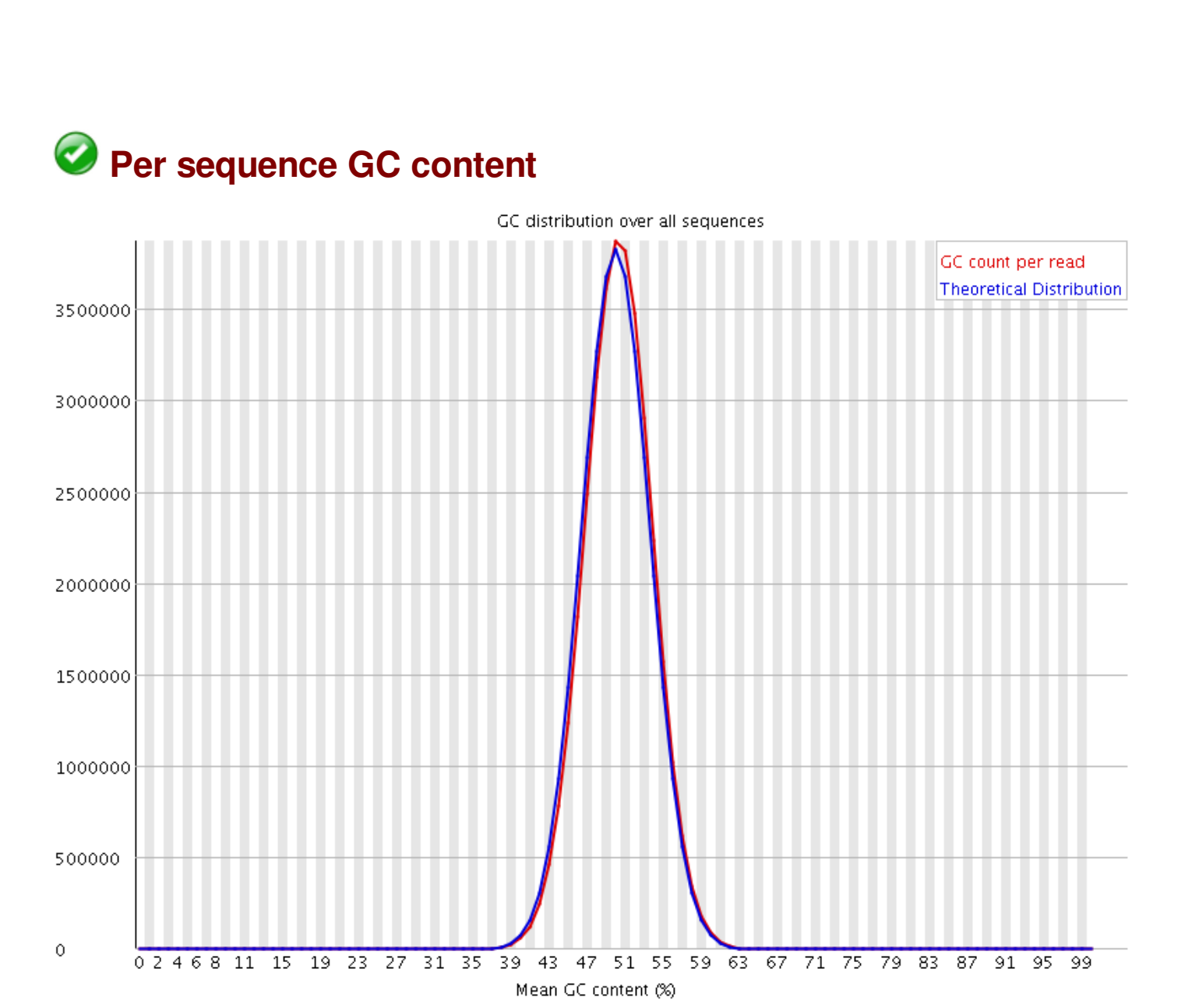
Per base sequence content



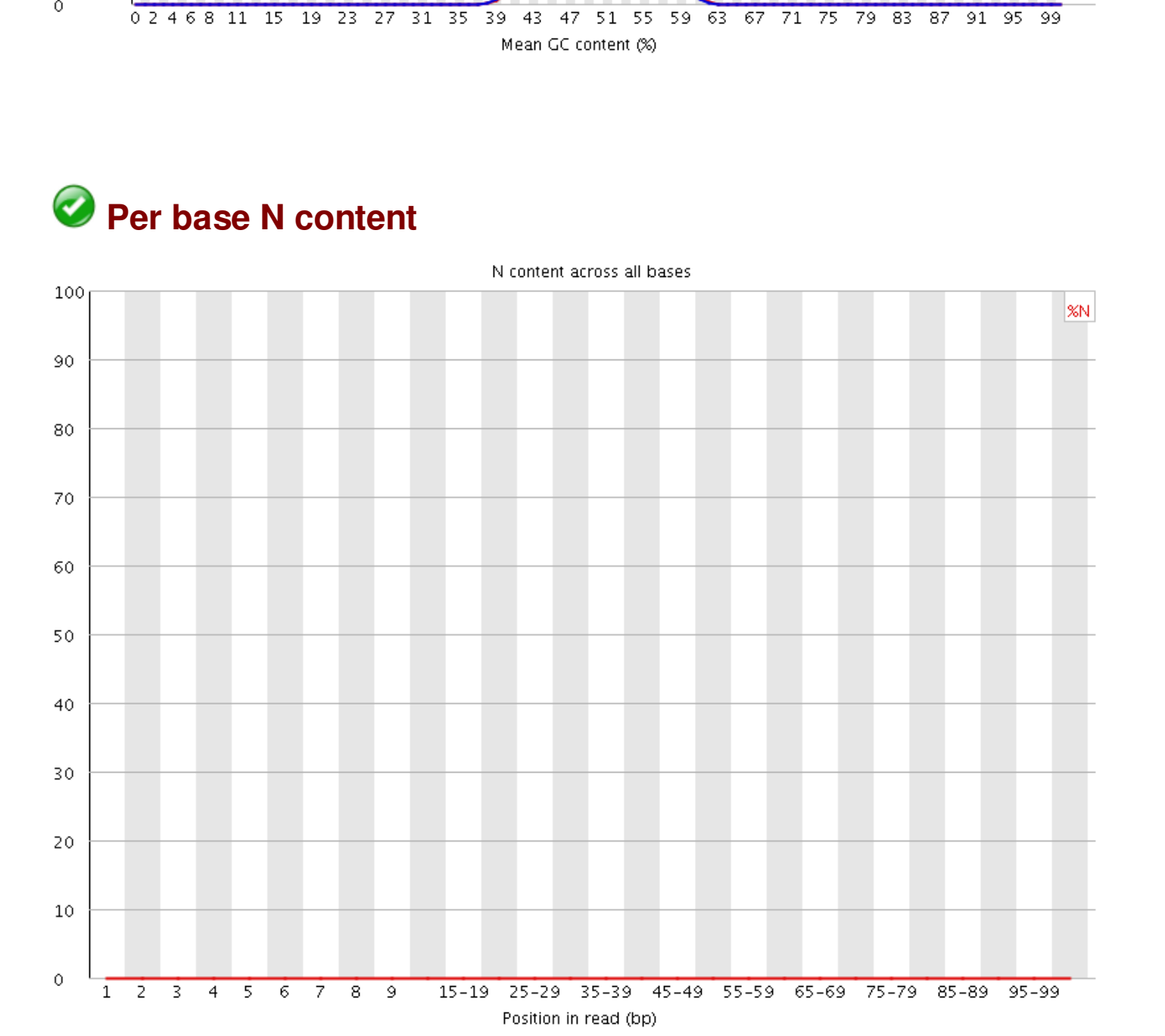
Per base GC content



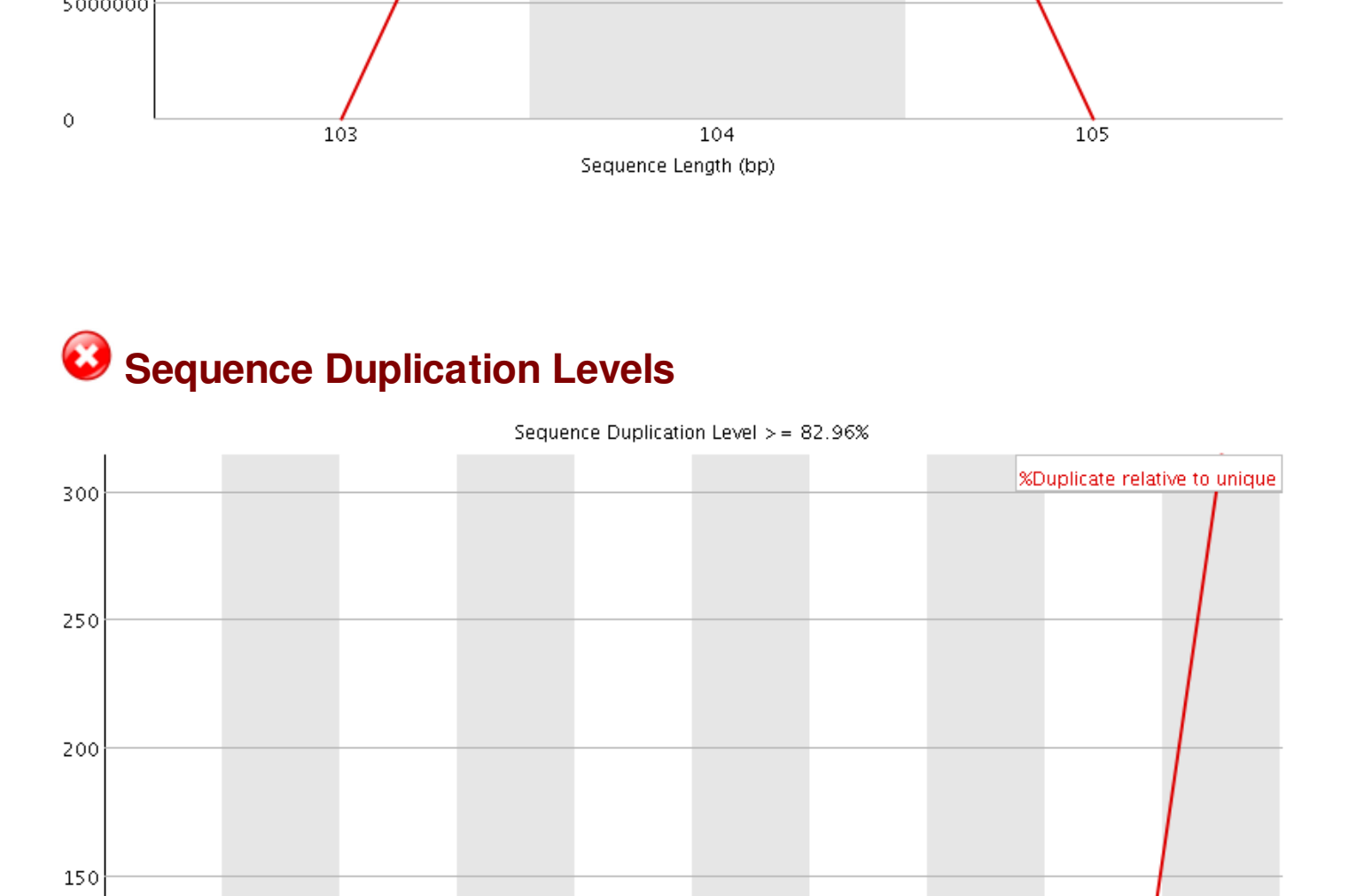
Per sequence GC content



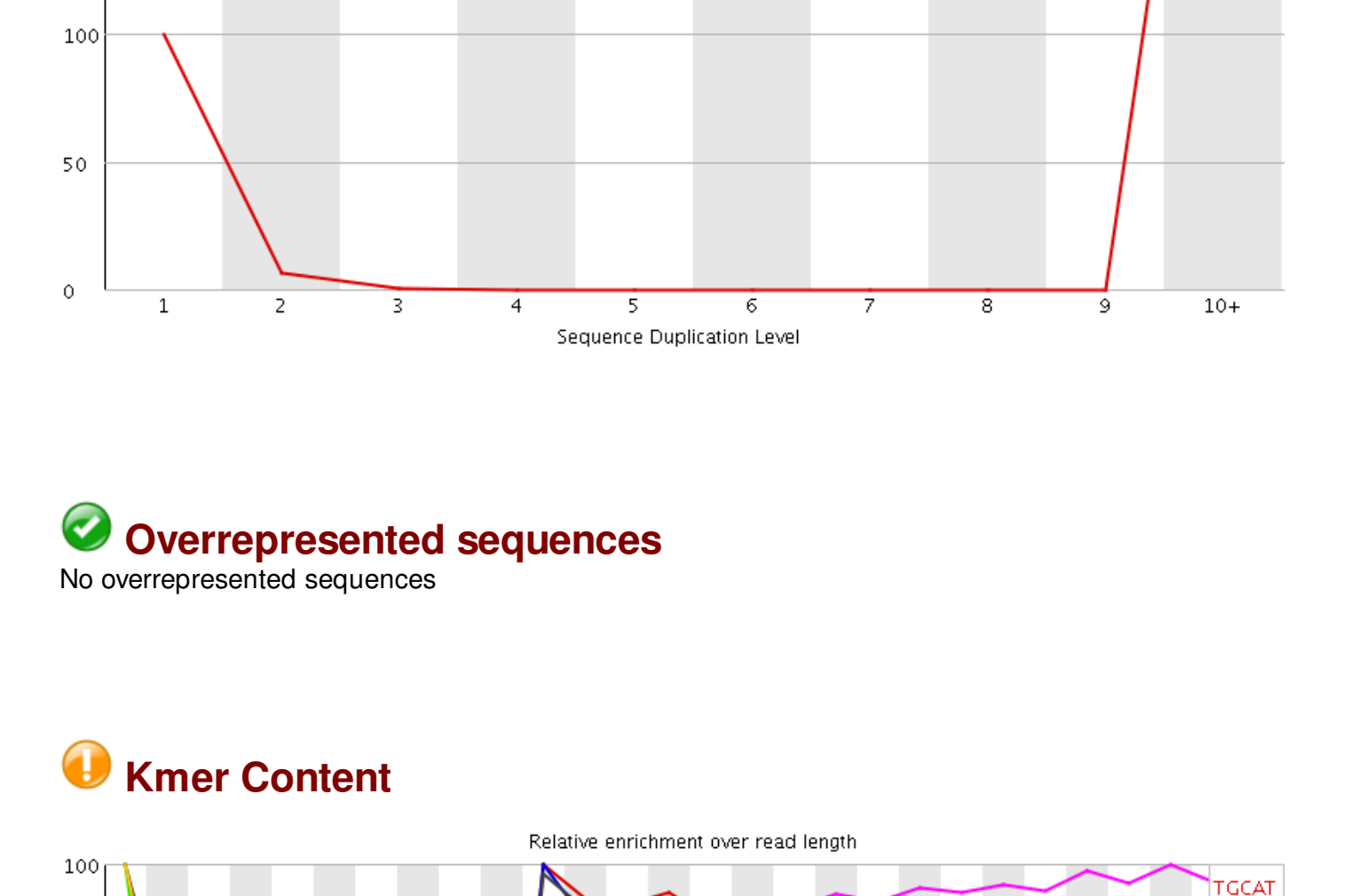
Per base N content



Sequence Length Distribution



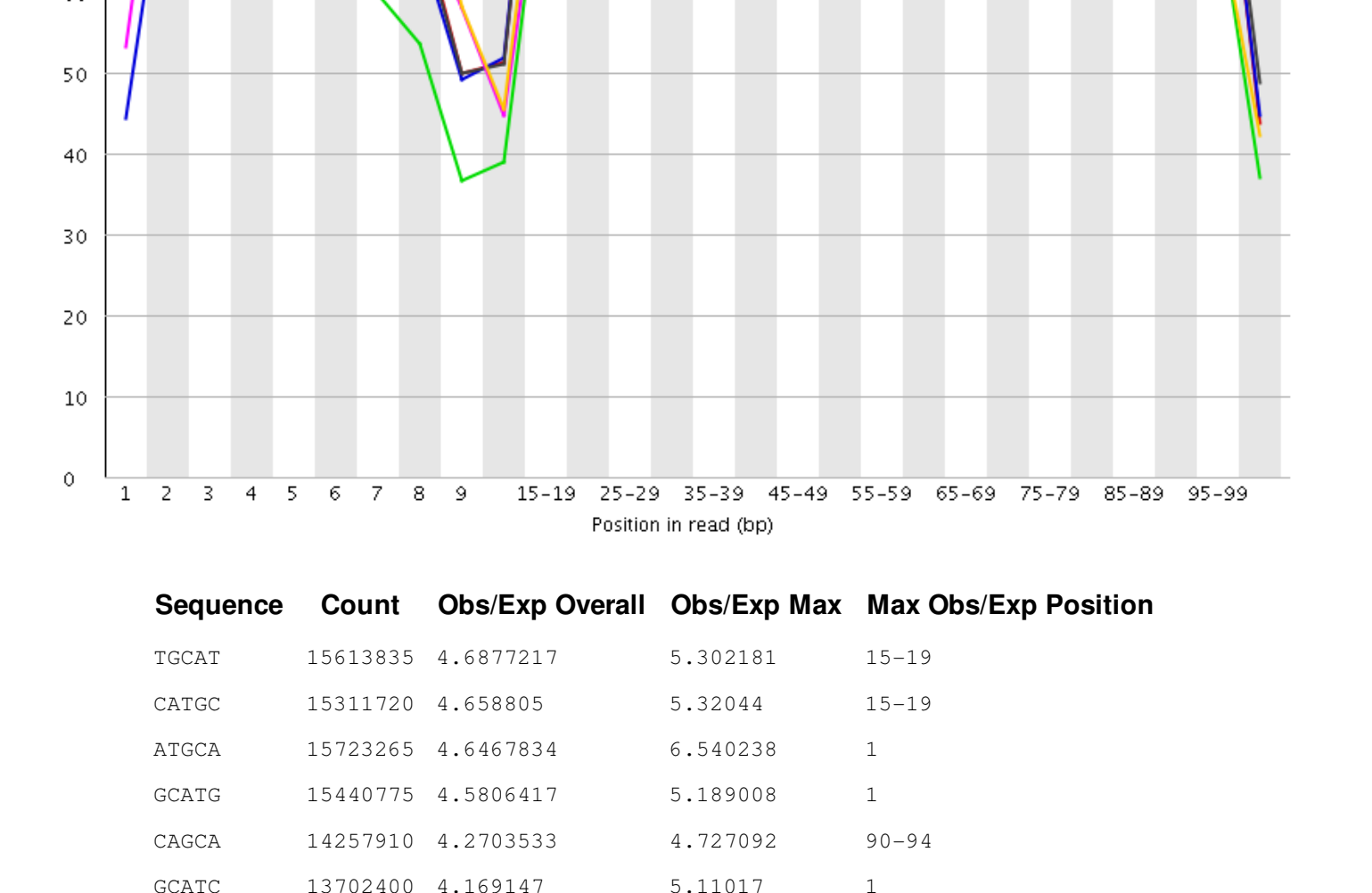
Sequence Duplication Levels



Overrepresented sequences

No overrepresented sequences

Kmer Content



Sequence	Count	Obs/Exp Overall	Obs/Exp Max	Max Obs/Exp Position
TGCAT	15613835	4.6877217	5.302181	15-19
CATGC	15311720	4.658805	5.32044	15-19
ATGCA	15723265	4.4467834	6.540238	1
GCATG	15440775	4.5806417	5.189008	1
CAGCA	14257910	4.2703533	4.727092	90-94
GCATC	13702400	4.163147	5.11017	1
CATCA	13672775	4.1443806	4.6072593	15-19
ATCAT	13836880	4.1384997	5.18818	1
ATGCT	13479140	4.046826	5.664456	1
GATGC	13573910	4.026819	4.4314094	40-44
TCATG	13404445	4.0244	4.42579	20-24
TGATG	13700495	4.0104694	4.3855915	25-29
TGCTG	13272820	4.000028	4.636728	1
CCTAC	13163400	3.9990633	22.822432	10-14
CTGCA	13125045	3.9934785	4.3507915	2
TACGT	13276360	3.9859455	22.784657	10-14
CATGA	13460050	3.9779227	4.3421555	20-24
AGCAT	13365965	3.9501173	4.321	20-24
TATGC	13149175	3.947761	7.9518943	1
ATGAT	13536815	3.947553	5.146608	1
TGCAG	13248875	3.9303954	4.299252	20-24
ACGTA	13262640	3.9195812	22.739258	10-14
GTACG	13135955	3.8968966	22.445755	10-14
ACATC	12805615	3.884565	4.21731	20-24
TGCAC	12629785	3.8427882	9.617908	100
CCCAT	12614615	3.8381734	5.097385	1
TCATA	12764335	3.8177102	8.469618	100
GCTGC	12461715	3.806063	4.6565857	1
GCACT	12508340	3.8058374	4.980494	1
CATGT	12676220	3.8057358	9.796993	100
GATGT	12976340	3.7984917	8.352408	100
TCGTA	12619020	3.788593	14.080935	100
TGTAC	12614020	3.787362	14.939585	100
ATGCG	12732020	3.7770665	9.04929	100
GCATA	12774540	3.7753303	9.170971	100
ATGTC	12561390	3.771291	4.4078736	1
CTATG	12512495	3.7566113	4.1141553	30-34
CGTAT	12511235	3.7562327	8.410614	10-14
CATAG	12696115	3.752153	4.0669823	15-19
GCAGC	12477755	3.7513878	5.4139557	1
TCGCA	12317435	3.747752	4.119753	20-24
GTGCA	12591285	3.7353156	4.0515785	10-14
AGCAC	12439300	3.7256656	8.147622	100
GTGCT	12360020	3.724938	4.00732	20-24
ACATG	12587520	3.7200592	4.148565	10-14
TATGA	12751780	3.718624	5.136688	1
CACGT	12154680	3.6982315	13.9250345	100
ATACG	12494835	3.6926675	13.934726	100
TGCGA	12411655	3.6820266	3.9893134	20-24
GACAT	12434795	3.6749234	3.9688184	30-34
AGTGC	12362870	3.667554	4.010015	20-24
GCGTA	12113485	3.593572	13.746721	100
AGTAC	12142295	3.5884793	12.926159	100
CTGCG	11748515	3.5882368	8.163651	100
CTACG	11723990	3.567188	13.980734	100
CGCAG	11832810	3.5574877	5.011808	1
GACGT	11882580	3.5250719	12.878546	100
AGATC	11909155	3.5195785	4.4370155	90-94
GTACT	11691430	3.5101032	11.329644	10-14
GTACA	11825435	3.494836	8.955261	10-14
GATCG	11775010	3.4931602	4.0574284	80-84
GTGTA	11903555	3.4844613	4.6855845	95-99
ACGTC	11447990	3.483211	10.903815	10-14
CACAT	11473025	3.4776106	3.820255	25-29
TACGA	11753790	3.4736626	11.038851	10-14
CGTAG	11614140	3.4454367	11.374617	10-14
TACGC	11230820	3.4171343	8.379055	10-14
CATCG	11198550	3.4073155	4.4967794	9
ATGTG	11614525	3.3998559	4.9317365	1
GCACA	11330970	3.3937123	5.5557137	1
GTCGT	11255075	3.3919406	7.5912647	10-14
TGTGC	11244600	3.3887842	3.680834	4
ACGTG	11355970	3.3688483	8.293049	10-14
CATAT	11148430	3.3344057	3.7747946	15-19
TCTGC	10748185	3.3222194	4.28572	1
GTATG	11340920	3.3197644	6.94853	10-14
CGATG	11170105	3.31371	4.941036	1
CGTCG	10840620	3.310947	7.317803	100
AGAGC	11337275	3.3107266	3.9533596	90-94
TGCTA	10977820	3.2958574	4.2452016	9
CATCT	10693990	3.2929516	3.7143486	9
ATGAC	11099075	3.2801712	4.512232	1
ACTGC	10780580	3.2801423	3.719684	2
ATATG	11238985	3.2774687	5.1587157	1
CTCAT	10641610	3.2768226	3.8514223	15-19
TACAT	10930020	3.2690814	7.3578057	10-14
GTCAT	10873355	3.2644937	3.5315301	30-34
TGCTC	10549555	3.2608235	3.593662	3
ATCGC	10709460	3.2585032	4.623581	1
GCAGT	10967095	3.2534852	5.1993957	1
GAGCG	11066425	3.2439158	3.8778791	9
AGCGT	10923165	3.240453	4.0701733	10-14
TCTCG	10476360	3.2381992	10.755919	100
TAGCA	10944745	3.2345612	5.4892764	1
GCTAT	10739250	3.2242317	4.6924706	1
CACTG	10582575	3.2198968	3.644505	2
TCACA	10580750	3.207151	3.5503004	5
GCAGA	10979530	3.2062573	5.1051173	1
CTAGT	10664500	3.2017896	5.763679	100
CATAC	10556045	3.1996627	4.5309644	100
TGTCA	10638095	3.1938624	3.5839972	6
CGTGT	10590335	3.1916084	4.8921156	1
TCGAC	10483225	3.1896677	5.845559	100
ATCTC	10357060	3.1892025	4.226954	95-99
GCGAT	10747795	3.188428	4.5264144	1
TAGTC	10591275	3.1798058	4.709892	1
TGCGC	10401960	3.1769712	3.5765839	3
GCGCA	10521125	3.1631348	4.8926973	1
TAGAT	10840405	3.1612363	4.0194335	1
GACTA	10690305	3.1593647	5.7225075	100
ATAGC	10687975	3.1586764	4.7818837	1
CAGTG	10594325	3.1428998	3.5547874	4
ATGAG	10905615	3.1424367	4.165508	1
CGACT	10320550	3.1401718	4.9040885	1
CGTGC	10272095	3.1373081	7.093483	10-14
TGACA	10599585	3.1325538	3.4877474	20-24
TGTGA	10664720	3.1218247	3.5042179	8
ACTAG	10556385	3.1197867	3.54185	9
AGTCG	10484180	3.1102242	5.520649	100
ATGTA	10664925	3.1100636	4.4889374	100
CTGTG	10319510	3.10999	3.8181572	2
CACAG	10381415	3.1093132	3.606858	2
ACGCA	10355145	3.1014452	6.8210006	10-14
CACAC	10071600	3.0938532	4.040316	9
TGTAG	10568645	3.0937011	3.9266407	95-99
ATCAC	10192780	3.0895534	4.74466	1
GTCGA	10408785	3.0878577	3.4547741	9
ATATC	10321540	3.0870895	4.2789726	1
GAGCA	10553670	3.0818973	3.7584498	15-19
AGATG	10677365	3.076667	3.8043027	15-19
TATCA	10262775	3.0695136	4.793198	1
GTAGA	10624410	3.0614078	7.2754827	10-14
GAGTG	10558765	3.054057	3.3609886	90-94
ACTAC	10060240	3.0493782	7.8325295	10-14
CTGTA	10147005	3.046423	4.05131	9
CACTC	9742415	3.0402584	3.3661404	2
CAGAT	10284845	3.039537	3.2990103	2
CGAGA	10397590	3.0363185	4.899002	1
TGCGT	10073130	3.035738	4.24626	100
TCAGC	9976075	3.0353603	3.4015882	80-84
AGAGT	10532125	3.0348163	8.404591	100
CTCGT	9809020	3.0319273	5.668236	10-14
GTCAC	9949735	3.0273457	3.8459954	9
TGTGT	10171050	3.0245965	4.348412	9
TCACG	9930820	3.021591	4.0864463	9
ATCGT	10051875	3.0178623	4.9327016	10-14
GTATC	10048965	3.016988	3.9238305	15-19
ACTGT	10035395	3.0129142	4.196879	9
TATCG	10035170	3.012847	4.2315593	1
TACTA	10070055	3.0118725	7.1586947	100
TCGTG	9990170	3.0107367	3.7574606	10-14
CGCTA	9878125	3.005558	4.948988	1
ACGAC	10016030	2.9998777	6.668892	100
CAGTA	10149795	2.999625	5.0562572	10-14
CACTA	9815415	2.9751692	7.386658	100
CGTCT	9615055	2.9719734	6.988253	10-14
GTAGT	10121720	2.9628747	6.9558167	10-14
TAGTA	10152790	2.9607167	7.8686004	10-14
GAGAC	10110995	2.9526267	8.456499	100
CTCTA	9586030	2.9517825	9.18412	100
TCTAC	9543760	2.9387667	5.7478952	10-14
TACTC	9504360	2.9266346	7.3410087	10-14
TGTCG	9672110	2.914883	6.7183466	100
CGACG	9690920	2.913537	7.2990775	10-14
ATCTA	9672735	2.8930373	9.42824	100
TACTG	9553125	2.8681233	5.0363903	1
ATAGT	9820795	2.8639016	6.7077346	100
ACTCG	9406970	2.8622022	8.541355	100
GTCTA	9531360	2.8615885	8.772109	100
TATGT	9618485	2.8483698	8.124417	100
CGAGT	9554395	2.8343954	7.960323	100
TAGAC	9566330	2.8271904	7.979701	100
TGAGT	9642115	2.8224826	8.820636	100
ACATA	9555280	2.8132317	7.767031	100
CAGAC	9374310	2.8076775	8.450829	100
CGCAC	9048060	2.7900007	7.6309896	100
ACGAG	9513750	2.778218	7.156014	10-14
GCGAC	9195080	2.7644646	6.053944	100
GAGTA	9558525	2.7542748	6.074305	10-14
GCTCG	8906215	2.7201402	9.191288	100
AGACG	9227885	2.6947393	5.811293	10-14
GTGCG	9008100	2.6824896	7.3733525	100
AAAAA	3009295	0.84711695	5.742332	100