# Conclusion

1. **Name of articles:**
   1) "Codes With Run-Length and GC-Content Constraints for DNA-Based Data Storage". (Wentu Song et al., August 22, 2018) {encoding theory}
   2) "Next-Generation Digital Information Storage in DNA". (George M. Church et al., August 16, 2012)
   3) "Forward Error Correction for DNA Data Storage". (Meinolf Blawat et al., August 13, 2016)
   4) "Robust Chemical Preservation of Digital Information on DNA in Silica with Error Correcting Codes". (R. N. Grass et al., February 4, 2015)
   5) "A DNA-Based Archival Storage System". (James Bornholt et al. April 2-6, 2016)
   6) "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA". (Nick Goldman et al. February 7, 2013)
   7) "DNA Fountain enables a robust and efficient storage architecture". (Yaniv and Dina, March 3, 2017)
   8) "Design of Capacity-Approaching Constrained Codes for DNA-Based Storage Systems". (Kees et al., February, 2018) {encoding theory}
   9) "A Rewritable, Random-Access DNA-Based Storage System". (S. M. Hossein… et al., September 18, 2015) {Rewritable}
2. **Time line:**
   2012 2) --> 2013 6) --> 2015 4) --> 2016 3) --> 2017 7) --> Feb 2018 8) --> Aug 2018 1)
3. **Background:**
   a) DNA sequence length cannot be too long. At most 250 nucleotides per sequence (oligo).
   b) Run-length at most 3
   c) GC-Content is close to 0.5
   d) High density
   e) Long-term storage
   f) Not restricted to a planar layer
   g) First demonstrated in 1988 by J. Davis
   h) Multiple copies are synthesized at the same time
   i) Swap error, insertion error and deletion error
   j) Swap error rate: [$6.0 * 10^{-4}$, $1.4 * 10^{-3}$], Insertion ad deletion error rate: [$1.0 * 10^{-3}$, $5.0 * 10^{-3}$]
   k) Bell shape function of oligo coverage
   l) Avoid self-reverse complementariness
   m) Progress in DNA storage is rapid with larger data being encoded and decoded successfully.
   n) Not all DNA sequences are created equal

4. **Background Reference**
   1) b), c)
   2) d), e), f), g)
   3) a), b), d), e), h), i), j), k), l)
   4) a), e)
   5) d), e), m)
   6) d), e), b):at most 2
   7) b), c), n)
   8) b)
5. **Idea**

1) Consider only GC-Content Constraint and Run-Length Constraint. Encode 2*n-1 bits binary sequences to n bits quaternary sequences. The encoding function/table depends on the last 3 quaternary bits of the last encoded sequence. Enumerate a 64 by $2^{2n-1}$ encoding table that satisfies run-length constraint and satisfies gc-content constraint as possible as it can.

3) Consider "Run-length limitation", "Insertion and deletion errors" and "Self-revers complementariness" constraints. Encode 8 bits binary data at a time. For the previous 6 bits, it uses the following table:

| Value | Nucleotide |
|---|---|
| 00 | A |
| 01 | C |
| 10 | G |
| 11 | T |

For the last 2 bits, it chooses one available option from the following table:

| Value | Opt1 | Opt2 | Opt3 | Opt4 |
|---|---|---|---|---|
| 00 | AA | CC | GG | TT |
| 01 | AC | CG | GT | TA |
| 10 | AG | CT | GA | TC |
| 11 | AT | CA | GC | TG |

By enumerating all possibilities, it can be proved that "for all possible 8-bit sequences x, there are at least 2 possible options for x." and "there exists an 8-bit sequence x such that there are 3 possible options for x.". Then the encoding table can be divided into 2 complete cluster A and B, and an incomplete cluster C. Encode each byte using cluster A and cluster B alternatively.

Finally, it uses multi-correction scheme to protect the encoded DNA sequence.