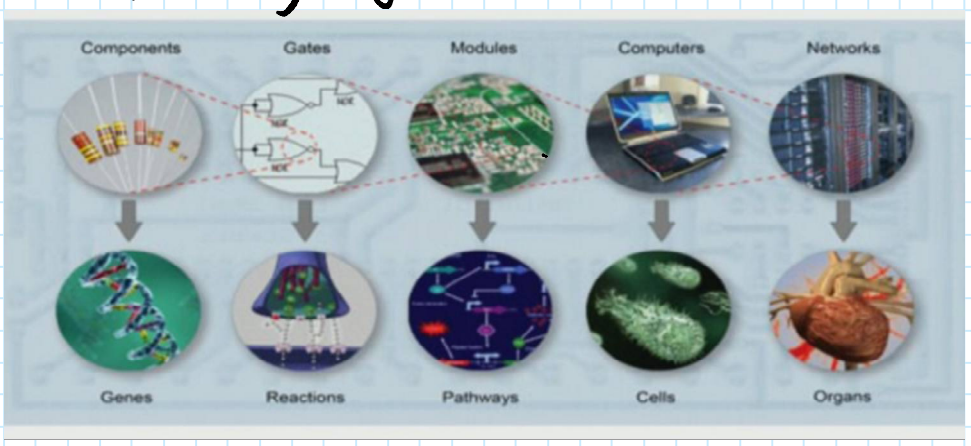
Chemical Computing

19 May 2017 13:57

Parallels between silicon and biological computing systems.

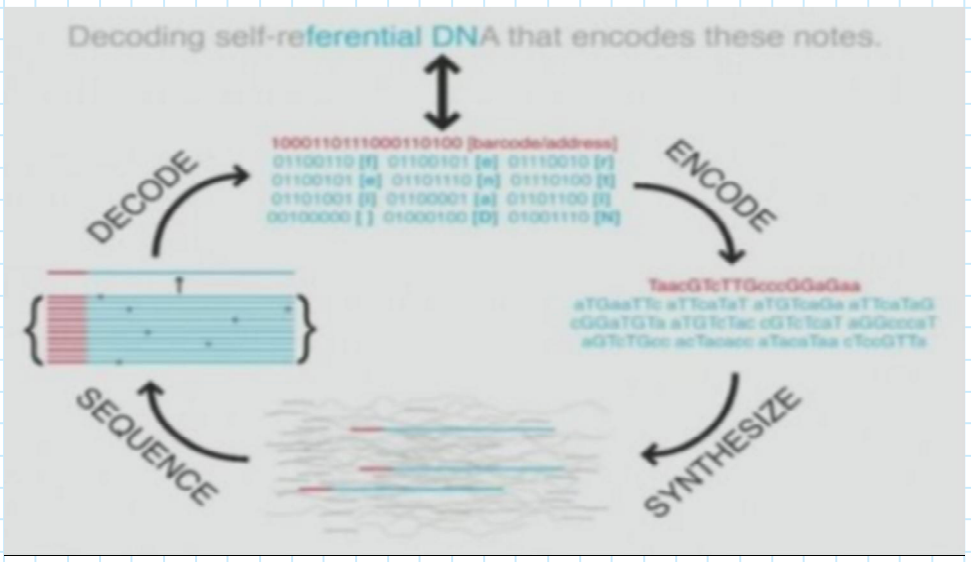


Information in DNA

DNA has a theoretical limit above 1 EB/mm^3 and has an observed half-life of 500 yrs in harsh envs. Optical disks have 100 GB/mm^3, durability of 30 years.

	Access Time	Durability
Flash	ms	~5 yrs
HDD	10s ms	~5 yrs
Tape	minutes	~15-30 yrs
DNA Storage	10s hrs	centuries

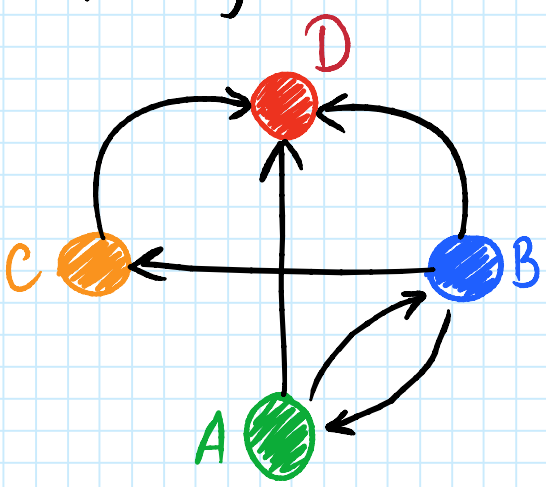
The compromise is access time v.s. storage capabilities.



We want to couple the pros of DNA storage capabilities with computational capabilities. The benefits of this would be highly parallel operations on high information density data. However, the techniques are currently expensive, error prone, application specific and involve lots of human intervention.

For example consider solving TSP with DNA

For example consider solving TSP with DNA computing



We assign each node a "DNA name"

ACTT GCAG
TCGG ACTG
CGCT ATGT
CCGA GCAA

These names are a two-part identifier, the first four symbols correspond to input and the second output. Thereby the edges of the graph are described by mixing and matching the outgoing and incoming descriptors. For example the edge $\overrightarrow{AB}$ is described by GCAGTCGG.

A→B	G	C	A	G	T	C	G	G
A→D	G	C	A	G	C	C	G	A
B→A	A	C	T	G	A	C	T	T
B→C	A	C	T	G	G	G	C	T
B→D	A	C	T	G	C	C	G	A
C→D	A	T	G	T	C	C	G	A

Put these strands (and a few complementary encodings), then take a 10^{14} molecules, water and ligase (DNA glue), salt and a few other cellular ingredients and mix them together. The mixture "performs" trillions of parallel computations (i.e chemical reactions), and the answer encoding the solution will appear somewhere within the mixture within a second.

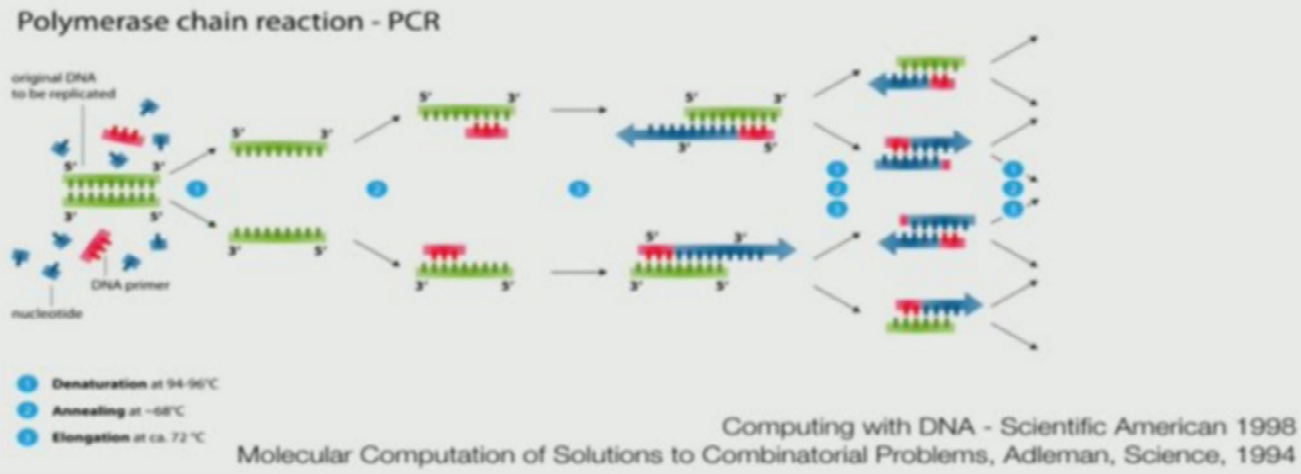
Traveling sales man

Polymerase chain reaction, or PCR, is used to replicate DNA molecules that start and end with the right primers corresponding to the right cities.

Use gel electrophoresis to separate out the strands with the right length.

Use iron-ball probes to check that all the cities are represented. Use a magnet to separate them out.

Use PCR and gel-electrophoresis to analyse the final molecule -> voila!



Computing with DNA - Scientific American 1998
Molecular Computation of Solutions to Combinatorial Problems, Adleman, Science, 1994

Post processing is the most intense step,

Post processing is the most intense step, i.e. extracting the answer.

Chess

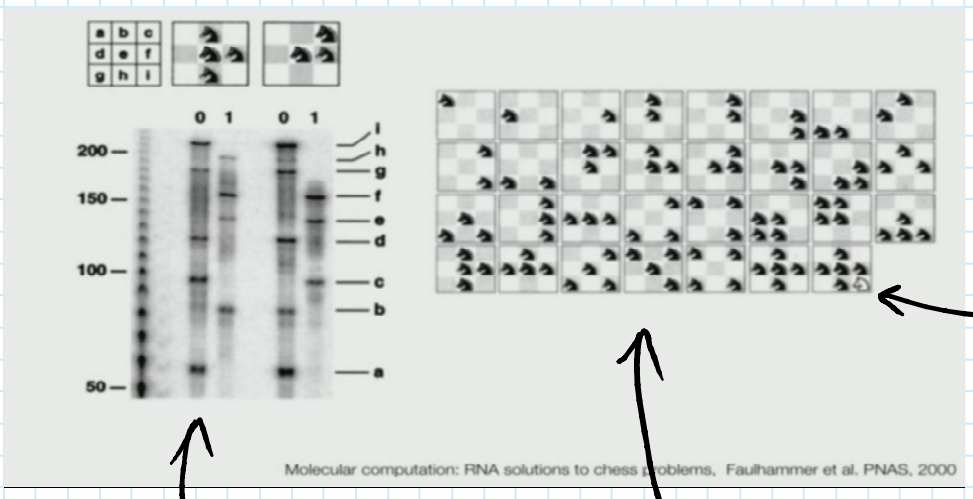
Knight problem: what configurations of knights can one place on an n x n chess board such that no knight is attacking any other knight on the board.

a	b	c
d	e	f
g	h	i

$$((\neg h \wedge \neg f) \vee \neg a) \wedge ((\neg g \wedge \neg i) \vee \neg b) \wedge ((\neg d \wedge \neg h) \vee \neg c) \wedge ((\neg c \wedge \neg i) \vee \neg d) \wedge ((\neg a \wedge \neg g) \vee \neg f).$$

Apply DeMorgan's, then knight problem is reduced to 3SAT ⇒ it's NP-Complete

The point is we encode the problem in DNA, then utilise massive parallism to find solutions. Therefore its a brute force approach to solving problems in NP, which then (by definition) can have their solutions verified in polynomial time. This last fact is important as DNA computing is fundamentally noisy and can make errors



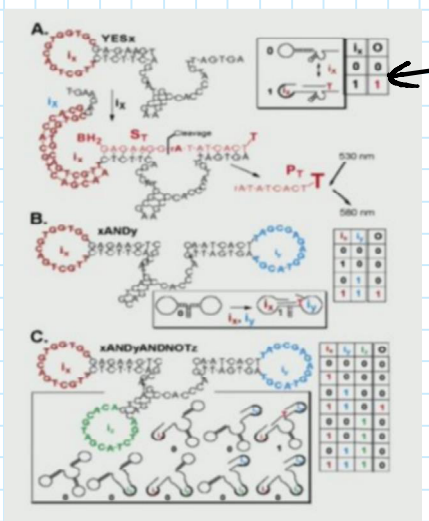
Molecular computation: RNA solutions to chess problems, Faulhammer et al. PNAS, 2000

Output of DNA computation

decoded solutions

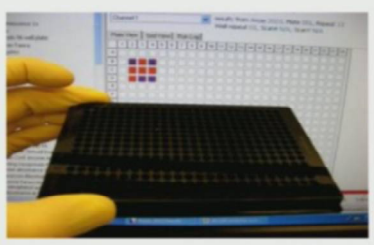
erroneous output of the computation, if this couldn't be verified in polynomial time the system wouldn't be as robust.

Reduction to SAT is the first step towards general purpose DNA-computing.
For example DNA logic gates,



becomes fluorescent upon matching to the correct input.

MAYA II (Molecular Array of YES and ANDNOT logic gates)



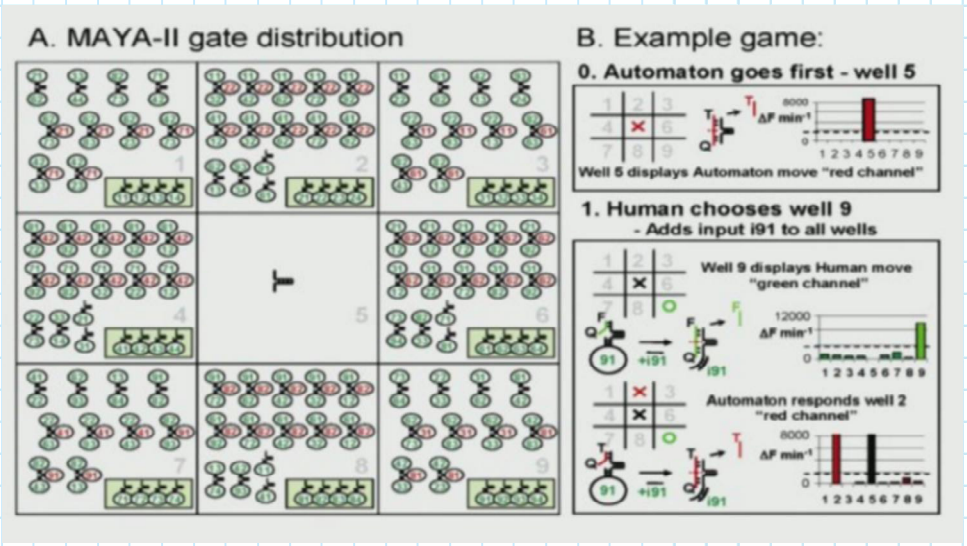
"Silicomimetic" approach to build molecular networks analogous to electrical engineering designs.

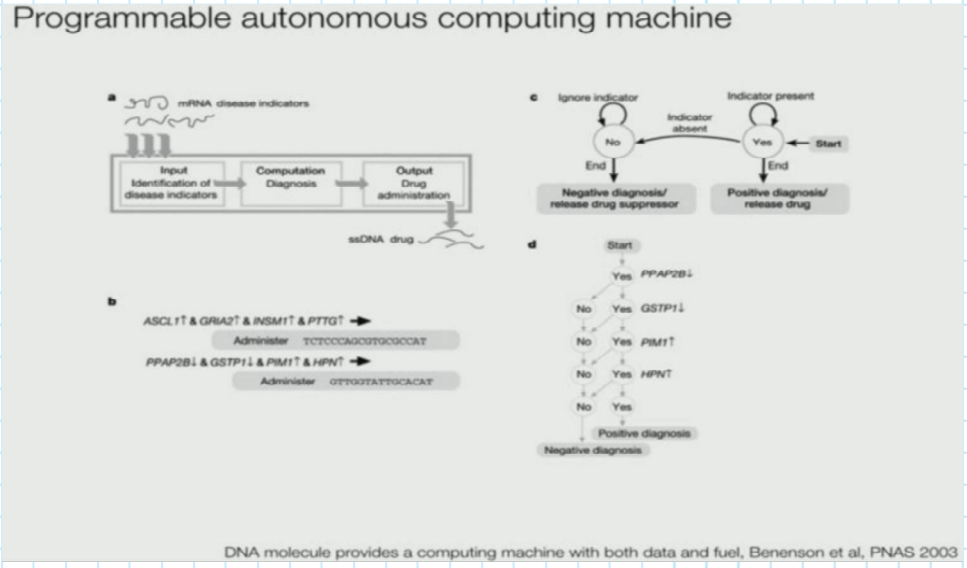
Networks consist of logic gates, which perform Boolean logical operations such as AND, NOT, and OR on one or more inputs to produce an output.

MAYA II uses more than 100 molecular logic gates to play tic-tac-toe.

Therefore a functionally complete set of logical

Therefore a functionally complete set of logical operators has been constructed. Example MAYA II "program" for playing tic-tac-toe.



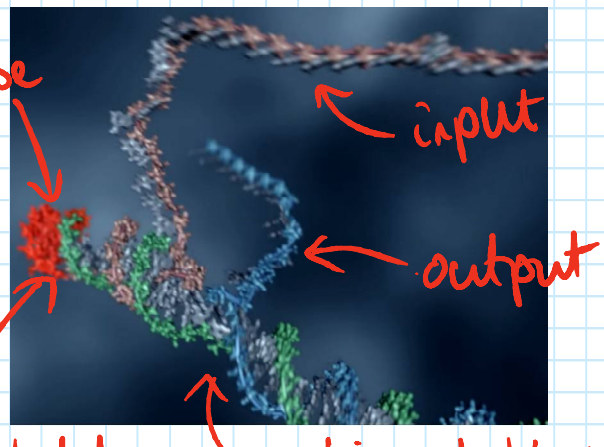


A huge motivator for DNA-computing is placing programs in the human body that respond to their chemical surroundings in predictable and helpful ways.

This could be used for sophisticated diagnostic applications or drug administration. The process of doing so could potentially be abstracted to the level of designing high level programs that are compiled to reaction networks.

DNA Strand Displacement Cascades:

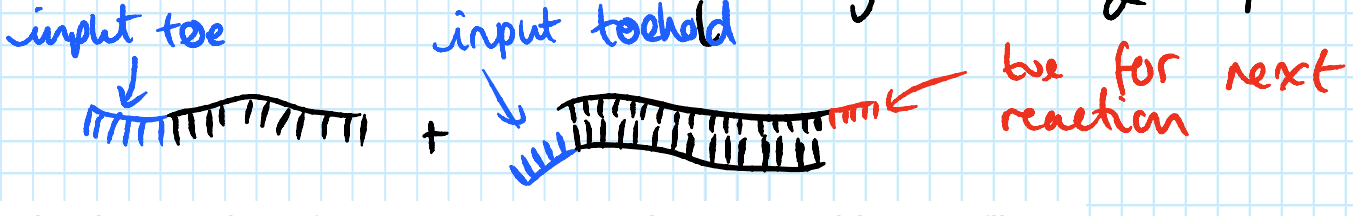
The idea is to have reactions relate to logical processes:



Microsoft Research's DNA Strand Displacement tool (DSD)

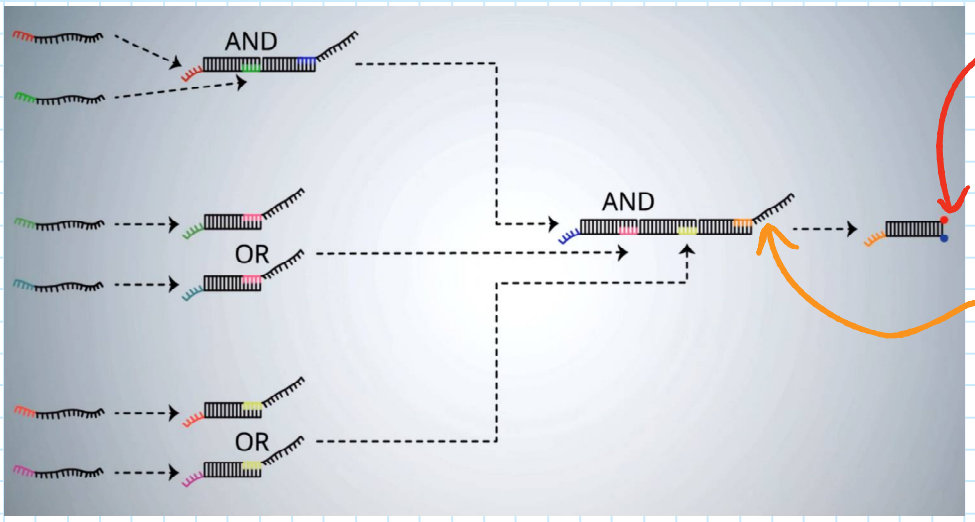
Basic reaction unit:

These reactions can be symbolically expressed



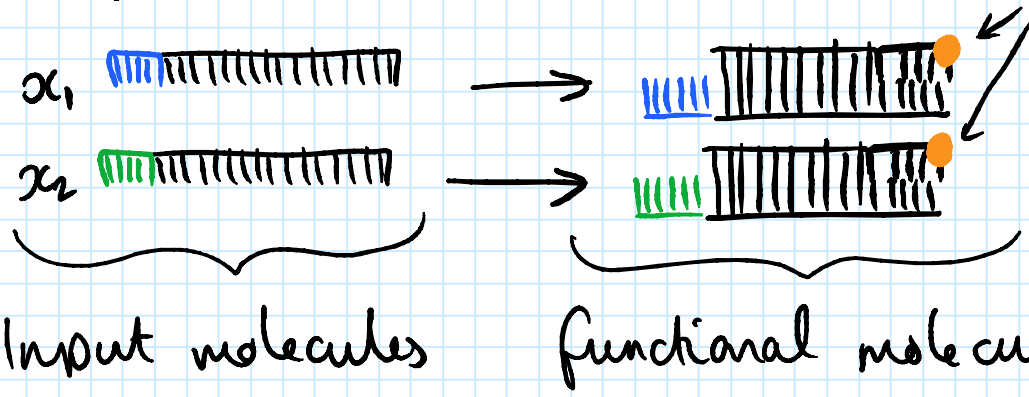
toe is free to go find next toe hold

Example network of these units, chaining revealing toeholds and releasing toes allows the creation of abitrarily complex expressions,



When this molecule is released it fluoresces, allowing output to be read. When the orange toe is released and binds to the orange toe, the fluorescence is activated.

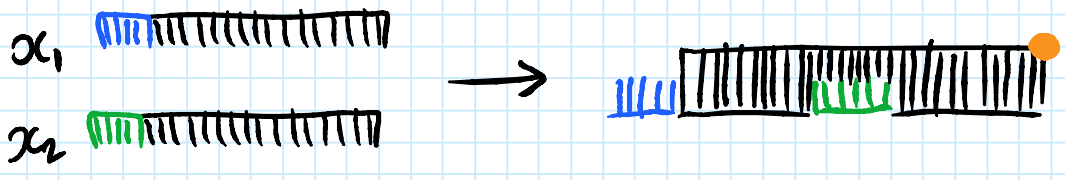
When designing reaction networks to compute functions we consider how the input to our system maps to brightness. For example if we have strands of type x_1 and x_2 in a mixture, and we want to measure $x_1 + x_2$. We can construct a network



Therefore provided there are enough functional molecules, the mixture will become more or less bright in accordance with how many input molecules that there are, i.e.

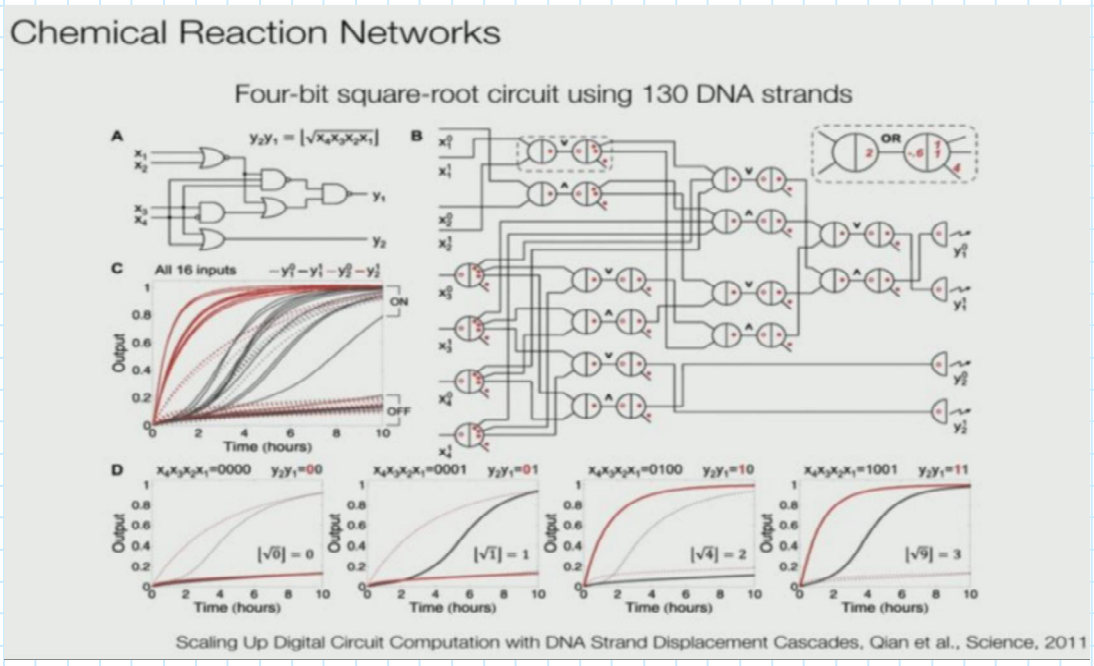
$$f(x_1, x_2) = x_1 + x_2$$

Functions of the form $f(x_1, \ldots, x_n) = \sum a_i x_i$ can be constructed similarly, where the functional molecules have fluorescent molecules of differing brightness in accordance with constant a_i values. Another example would be $f(x_1, x_2) = \min(x_1, x_2)$



$f(x_1) = \frac{1}{2} x_1$

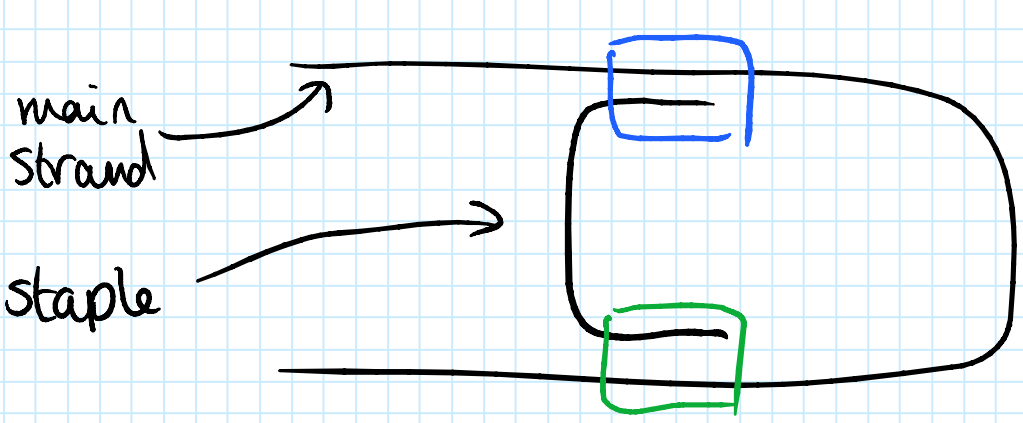
State-of-the-art in network design:



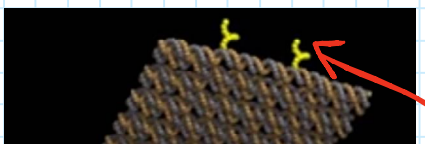
Currently DNA computers are limited to ~20 gates, as each gate increases the chance for error at an exponential rate.

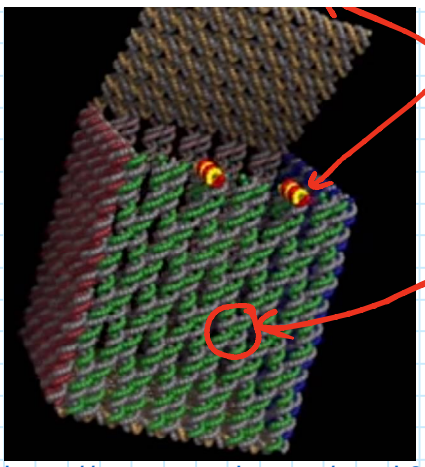
DNA Origami

An alternative DNA computing method (that can be used in conjunction with previous methods), is DNA origami. This idea is born from the fact DNA strands form double helices by matching base pairs in the same way each time. By designing "staples" we can bind strands to themselves



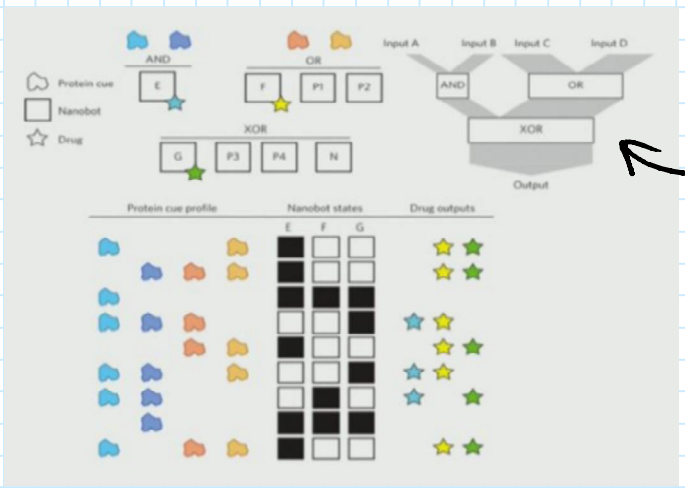
The parallel, colour-coded regions of the main strand and the staple have matching sequences, so they bind and hold the strand in the folded position. Adding more staples makes it more rigid, and adding more folds allows for creating arbitrarily complex structures. For example the box structure to the left is constructed by stapling strands into lattices that





https://www.youtube.com/watch?v=Trg2__Lgnc0

strands into lattices that form sheets. Additionally a lock structure is put in place by implementing toehold-activated cascades. Thereby constructing a nanobot capable of transporting drugs and releasing in accordance with specific conditions.

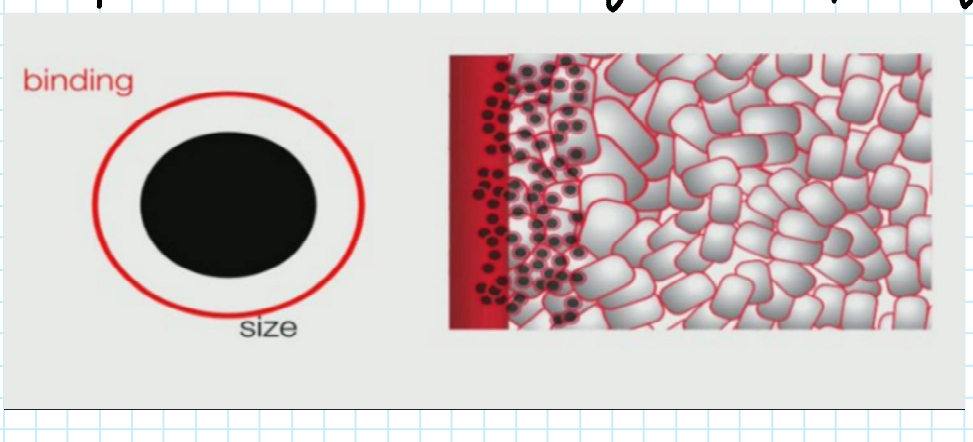


example logic for drug release

A problem is creating systems that are robust to being within living systems.

Nanobots for Cancer Treatment/Diagnostic:

- Nanobot construction naturally generates swarms
- Nanobots could be small enough to navigate tumors
- Tuning design parameters is difficult.
- Alternatively to designing logical systems (such as displacement cascades), behaviour can be implemented through morphology



for example tuning the size/stickiness of these particles for tumor identification.