An experimental realization of a universal computer
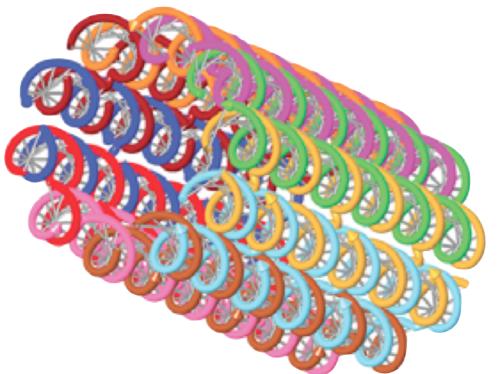
Nicolas Schabanel

CNRS

LIP & IXXI - ÉNS de Lyon

Slides mainly borrowed from Damien Woods et al (Nature 2019)
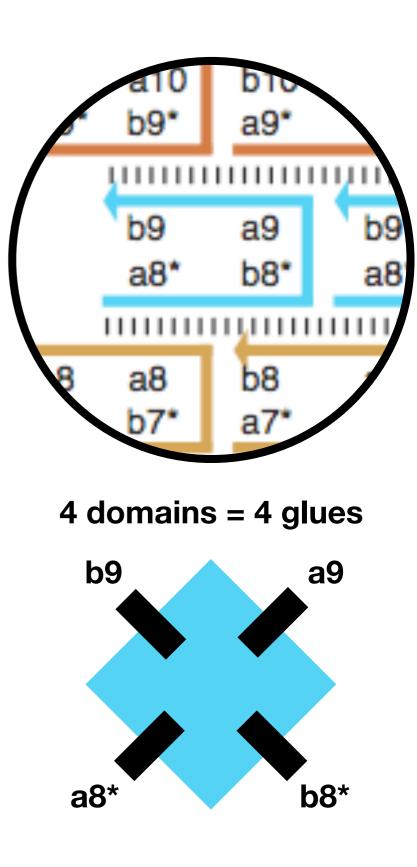
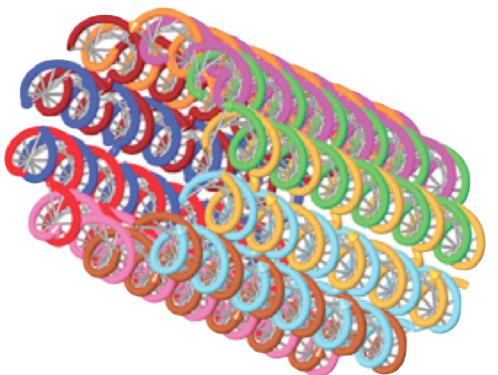# Single Stranded Tiles Nanotubes

Single stranded Nanotubes

10-helix nanotube schematic,
Yin et al. '08

4 domains = 4 glues

3

Single stranded Nanotubes

10-helix nanotube schematic,
Yin et al. '08

4 domains = 4 glues

3

Growing them

Seed
=
DNA Origami
encoding
an input

STT
assembly
=
Computation

○ ○ = 0
● ● = 1

U2

wire 1

U3

wire 2

U4

wire 3

U5

wire 4

U6

wire 5

4

Seeded growth: barrier to nucleation at [tile]=100nM

Experiments give a (narrow) temperature range at which we make good-quality, long, nanotubes!

100nM

Lower concentration
=> bigger barrier to nucleation!
2.5°C gap!

~2.5°C

1µM

Higher concentration
=> longer nanotubes!

8-helix. 1µM. 90 -> 56, hold for 1 day. Then room temp to 58.9 C, and hold for 1 more day. 10µM scale bar

8 Helix. 53.3C (Too cold: nanotubes & blobs)

Too cold: nanotubes along with blobs & nanotube tangles

Just right: long nanotubes

Too hot: free tiles do not bind to each other

Too cold: nanotubes along with blobs & nanotube tangles

Just right: long nanotubes

Too hot: free tiles do not bind to each other

8-helix. 100nM. 53.0 C | 12-helix. 100nM. 52.2 C | 14-helix. 100nM. 52.2 C | 16-helix. 100nM. 52.2 C

8-helix. 1µM. 57.0 C | 10-helix. 1µM. 57.0 C | 12-helix. 1µM. 56.2 C | 14-helix. 1µM. 56.2 C | 16-helix. 1µM. 56.2 C
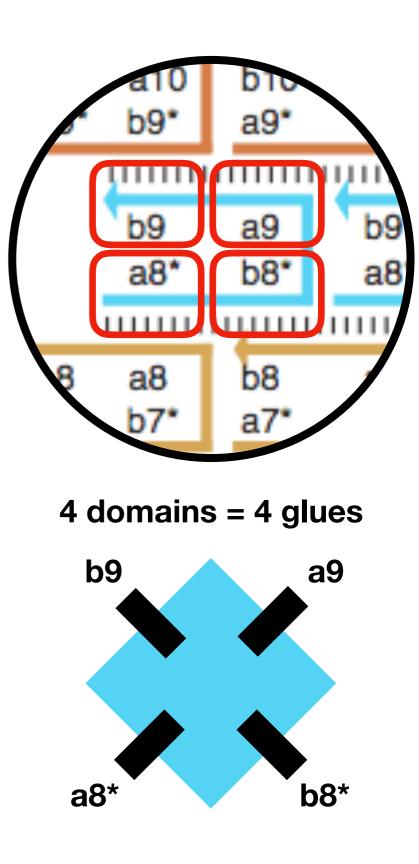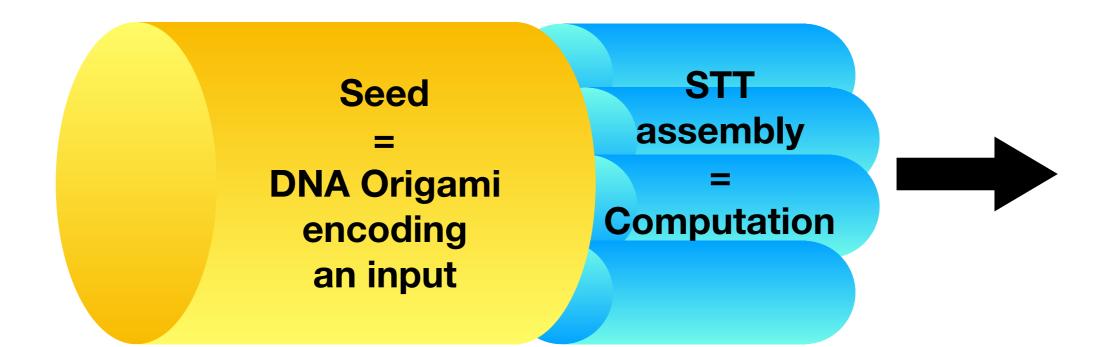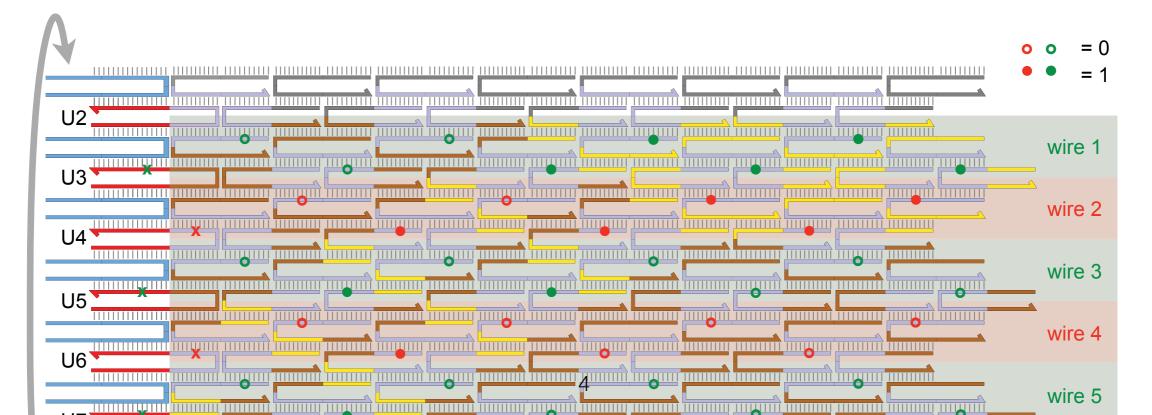
All scale bars 10µm
Each datapoint is a separate ~24 hour temperature hold experiment
Fluorescence microscopy images: cy3 label
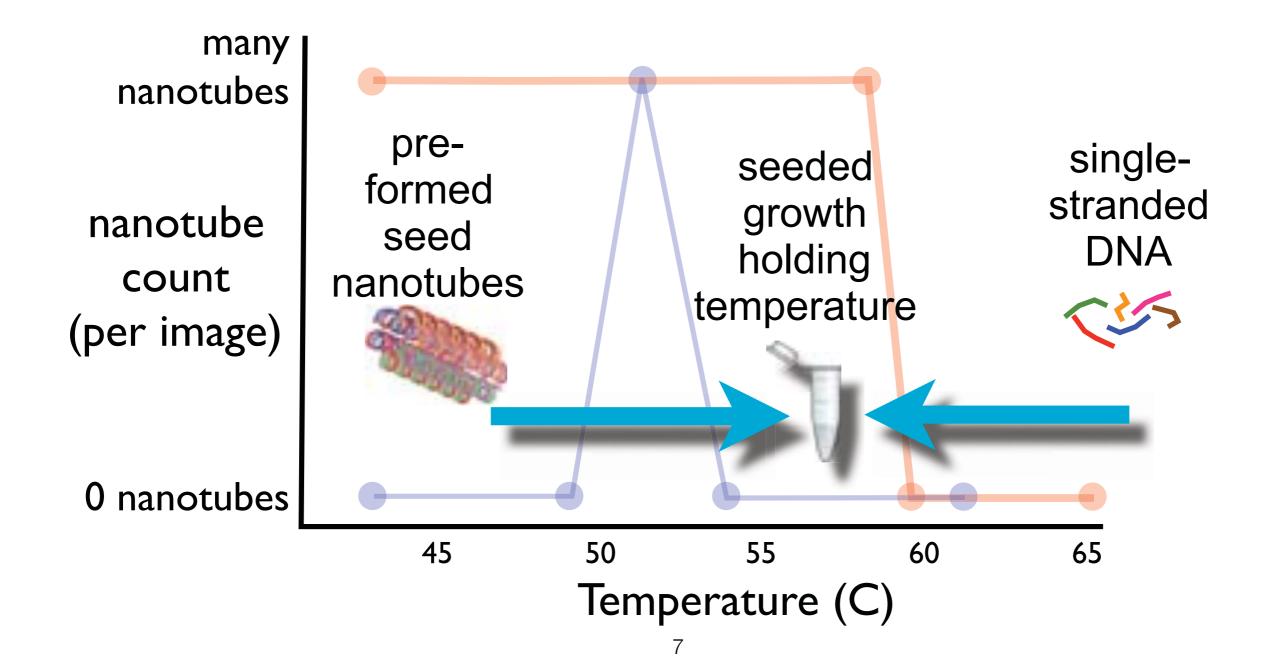Error bars show SEM for *n*=5 experiments for blue, and *n*=2 for red
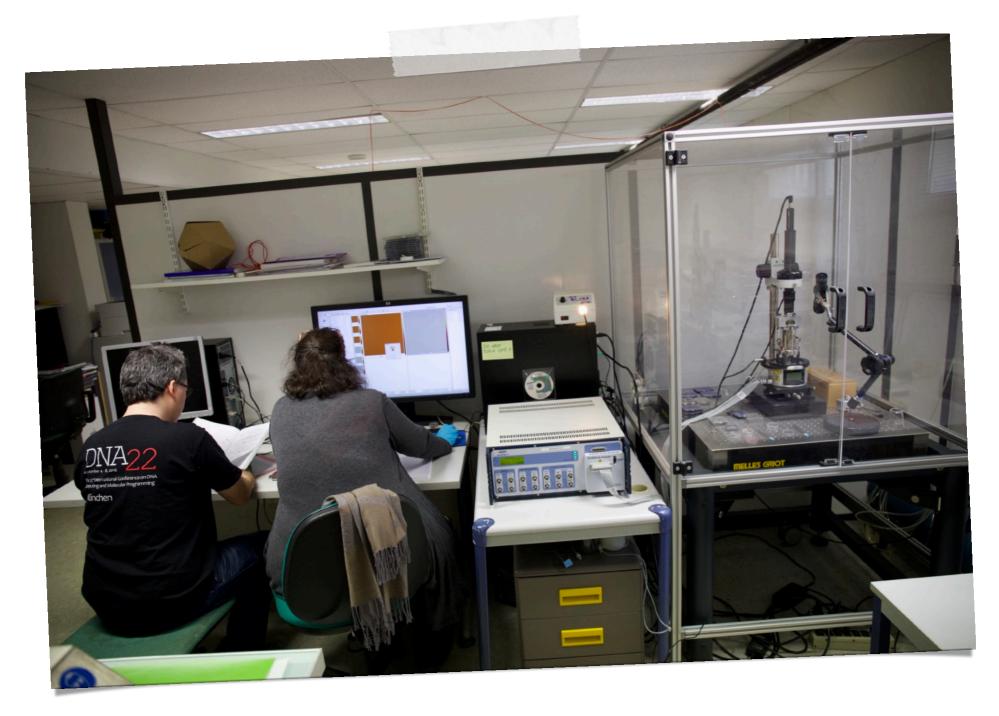
5

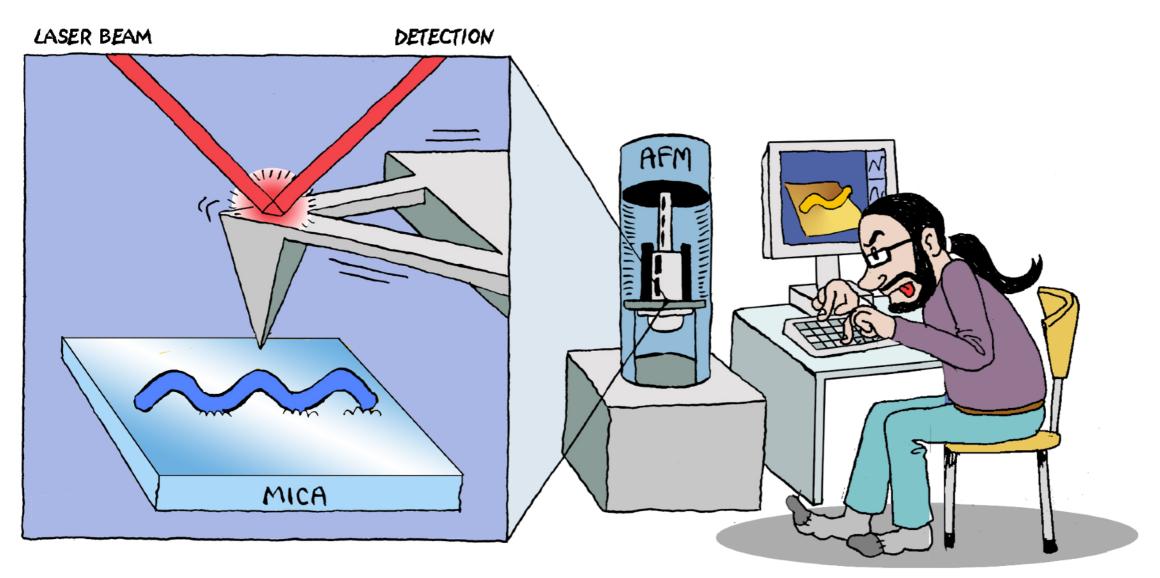# Seeded growth: barrier to nucleation at [tile]=100nM

Growth from seed only

Growth from seed → Seed →

[tile] = 100nM

Lower concentration => bigger barrier to nucleation! 2.5°C gap!

Everything sticks together

~2.5°C

Everything melts

End-to-end joining

seeds: 8-helix nanotubes, Alexa647 labelled
growth from seed: 8-helix tiles, Cy3 labelled
scale bar: 10μm. 100nM tile concentration

Controls: *0 seed nanotubes => 0 nanotubes/image*

Take pre-formed seed nanotubes, at room temperature and heat to temperature *T*

Single stranded

Many nanotubes (per image)

0 nanotubes (per image)

Temperature (C)

We work at this temperature

Too cold: nanotubes along with blobs & nanotube tangles

Just right: long nanotubes

Too hot: free tiles do not bind to each other

8-helix. 100nM. 53.0 C   12-helix. 100nM. 52.2 C   14-helix. 100nM. 52.2 C   16-helix. 100nM. 52.2 C

Scale bars 10μm. ~24 hour temperature hold experiments. cy3 label

Damien Woods

Joy Hui   6

Seeded growth only

Imaging the results
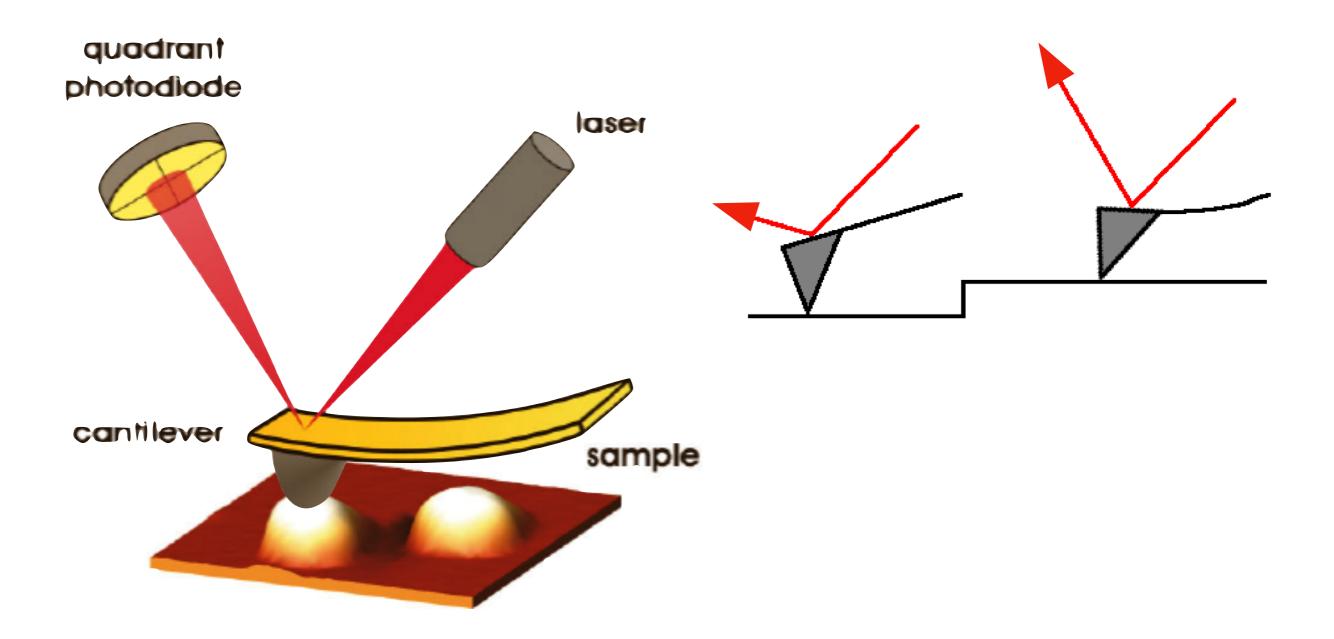
Principle of Atomic Force Microscopy
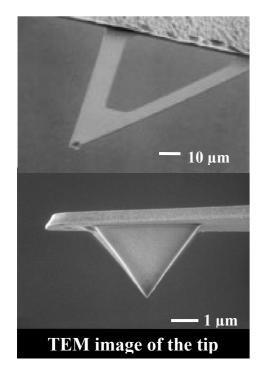
The microscope works by scanning the surface with a sharp probe and gently touching the DNAs that arrange on the mica.

9
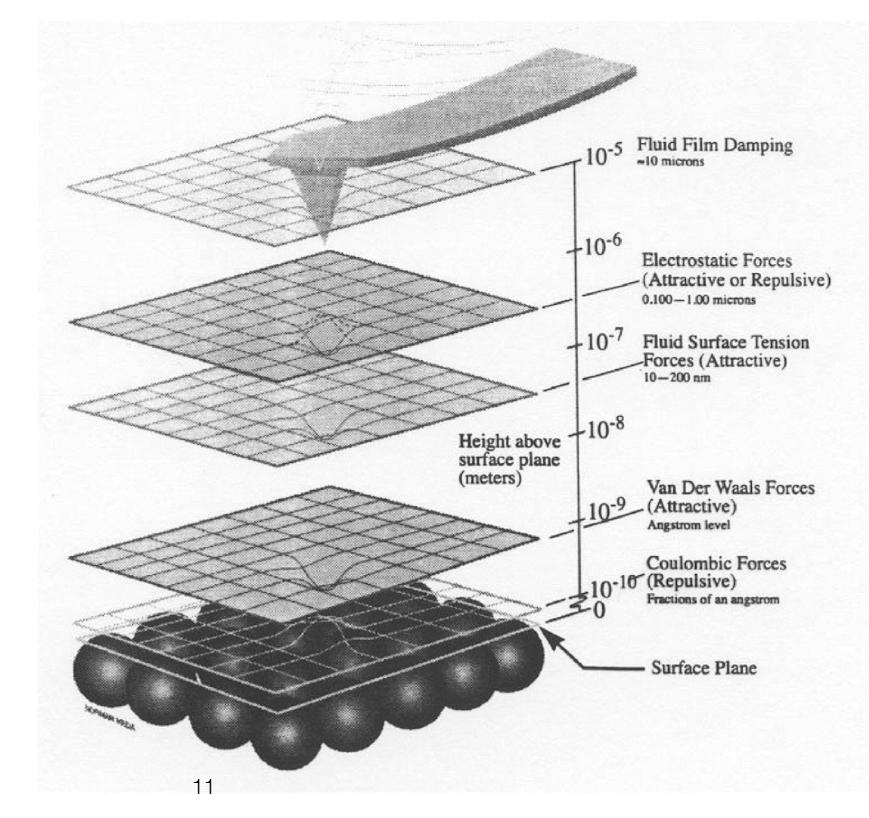
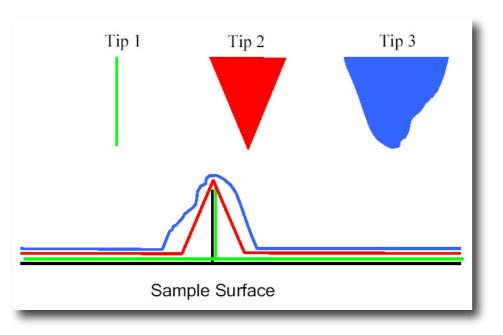# Laser deflection
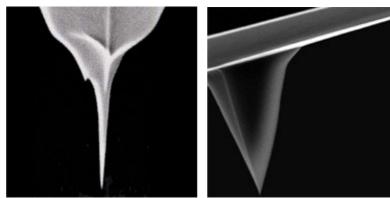
The forces involved in AFM

— 10 µm

— 1 µm

TEM image of the tip

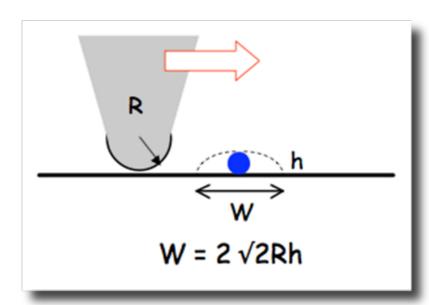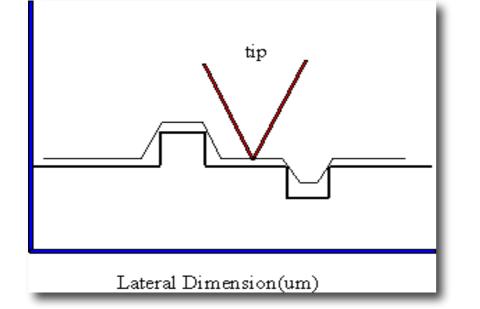They are interaction forces between the atoms of the end of the tip and the atoms on the sample surface.

10^{-5} Fluid Film Damping
≈10 microns

10^{-6} Electrostatic Forces
(Attractive or Repulsive)
0.100—1.00 microns

10^{-7} Fluid Surface Tension
Forces (Attractive)
10—200 nm

Height above
surface plane
(meters) 10^{-8}

Van Der Waals Forces
(Attractive)
10^{-9} Angstrom level

Coulombic Forces
(Repulsive)
10^{-10} Fractions of an angstrom
0

Surface Plane

Tip convolution

Sample Surface

Tip 1   Tip 2   Tip 3

Lateral Dimension(um)

tip

Tip radius 2-20 nm

$$W = w + 2 \sqrt{h(2R-h)}$$

$$W = 2 \sqrt{2Rh}$$
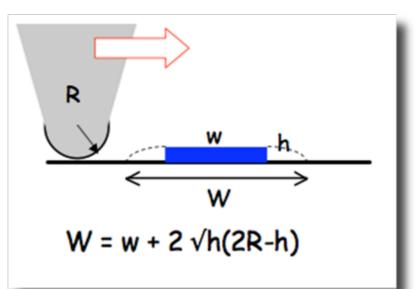
High resolution imaging

Fig. 1. STM and AFM imaging of pentacene on Cu(111). (**A**) Ball-and-stick model of the pentacene molecule. (**B**) Constant-current STM and (**C** and **D**) constant-height AFM images of pentacene acquired with a CO-modified tip. Imaging parameters are as follows: (B) set point I = 110 pA, V = 170 mV; (C) tip height z = −0.1 Å [with respect to the STM set point above Cu(111)], oscillation amplitude A = 0.2 Å; and (D) z = 0.0 Å, A = 0.8 Å. The asymmetry in the molecular imaging in (D) (showing a "shadow" only on the left side of the molecules) is probably caused by asymmetric adsorption geometry of the CO molecule at the tip apex.

About AFM scale

… how to shake the Mont Blanc over
little men heads without crushing them

Oscillation : 20 m

Cantilever 100 km

Mont Blanc :
4807 m

Man: 2 m

expanding to our scale : multiply all by 10^9

Marking 0s and 1s
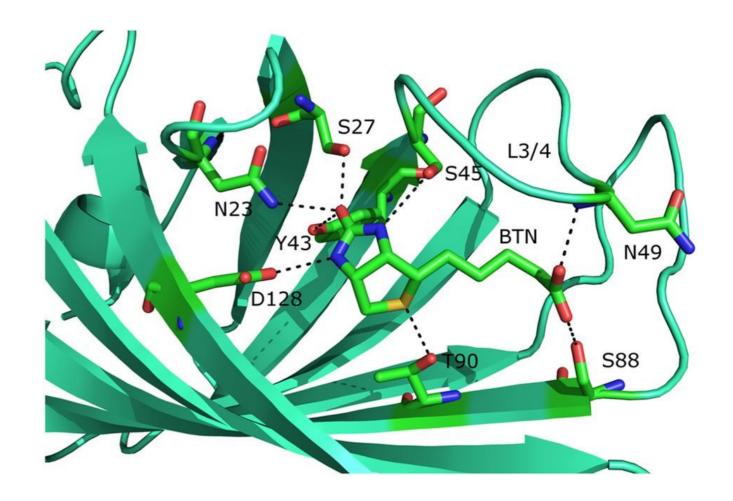
Streptavidin-biotin marker

Streptavidin : a "huge blob"
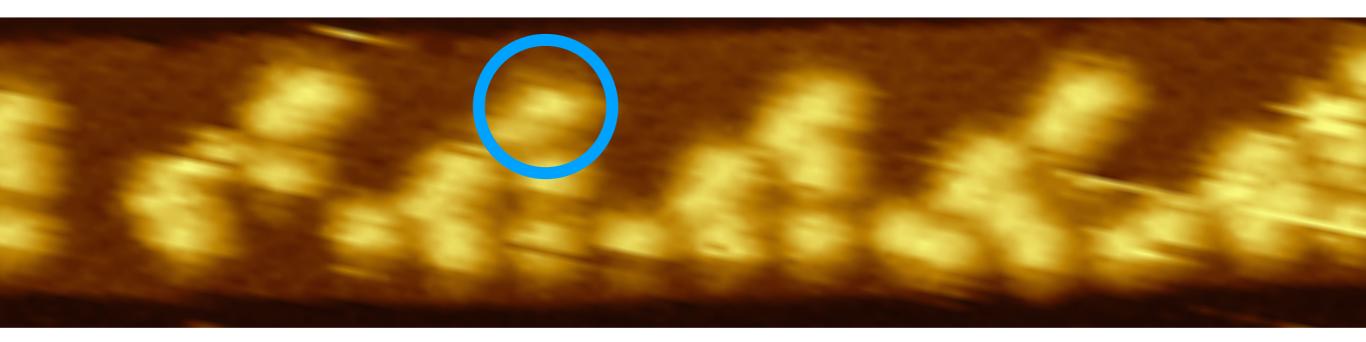
Biotin can easily be attached to DNA strand at order

Together they make one of the strongest non-covalent bond
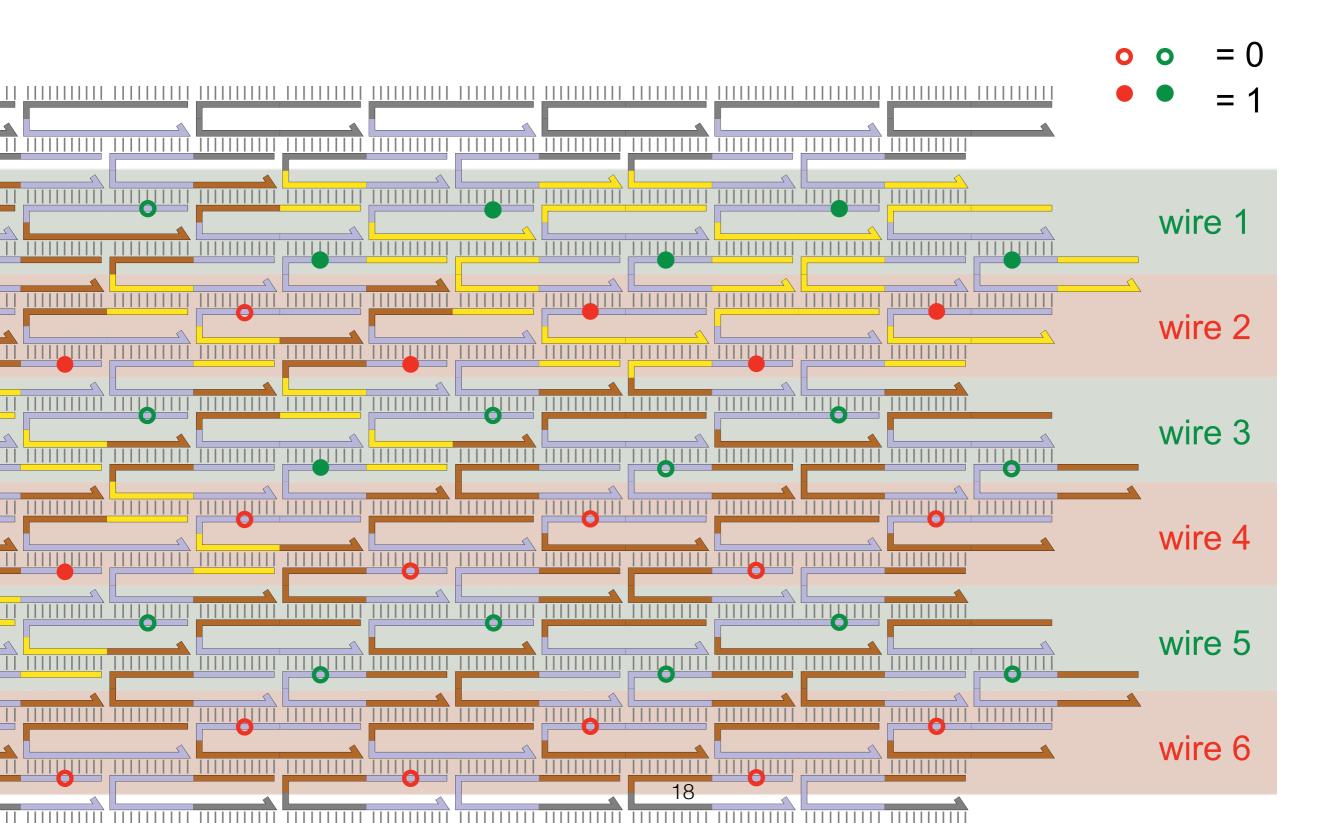
Streptavidin-biotin marks

We can order single DNA strand with biotin attached
(the tiles encoding a 1!)
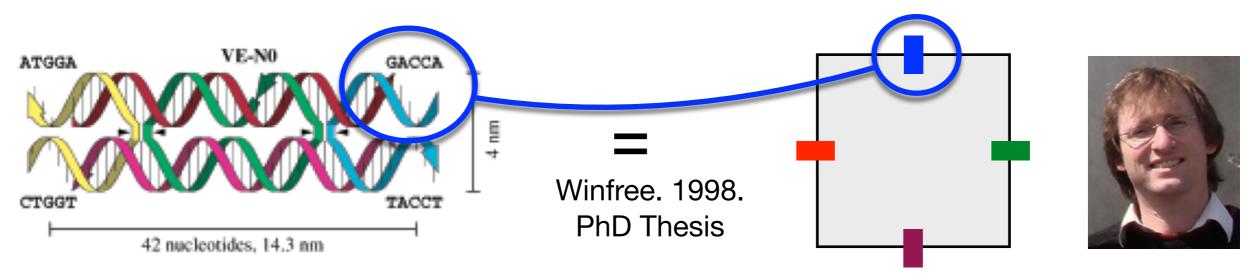
When added to the solution while imaging,
Streptavidin attaches to biotin,
marking the corresponding single stranded tiles

Streptavidin-biotin marks

wire 1

wire 2

wire 3

wire 4

wire 5

wire 6

○ ○ = 0

● ● = 1

18
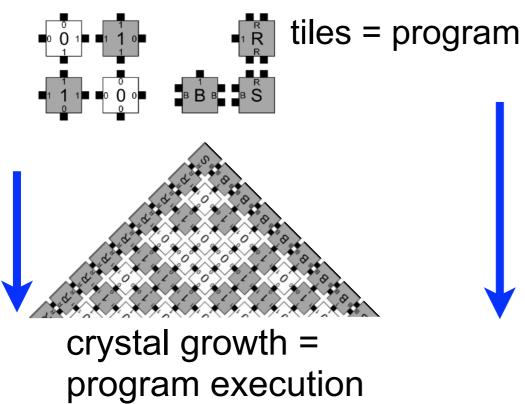
kTAM model for algorithmic assembly
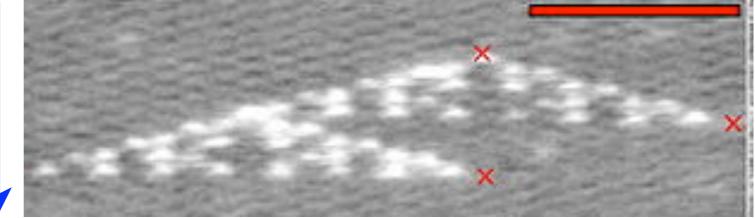
Algorithmic self-assembly

VE-N0

ATGGA GACCA

CTGGT TACCT

42 nucleotides, 14.3 nm

4 nm

=

Winfree. 1998.
PhD Thesis

Erik Winfree had the idea that a growing lattice of DNA tiles could run a computer program, like Wang tiles or a CA

tiles = program

crystal growth =
program execution
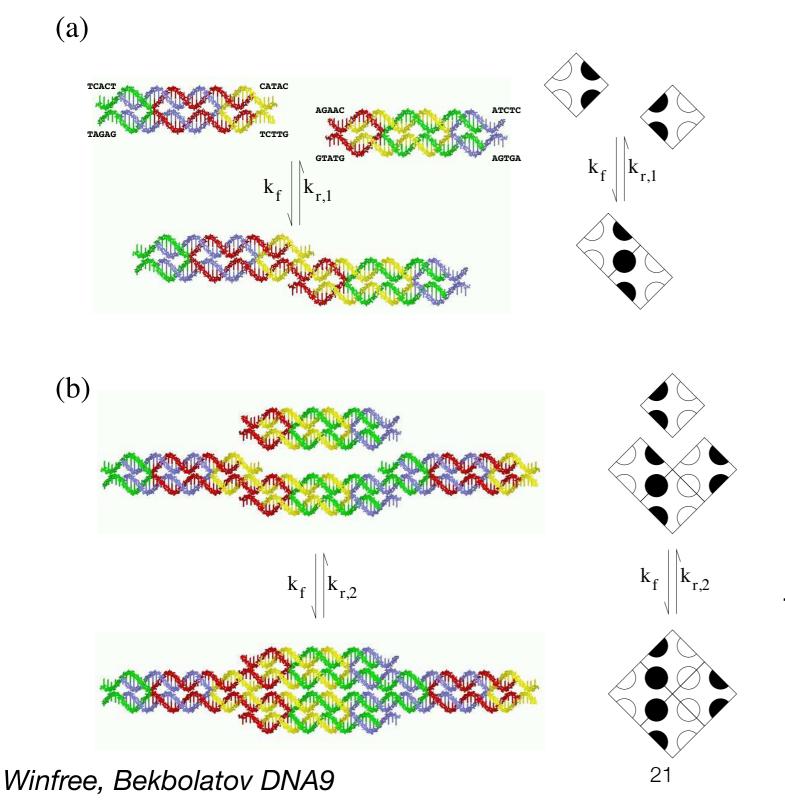
Rothemund, Papadakis, Winfree 2004

Thermodynamical model

(a)

(b)

Attachement rate

$=$
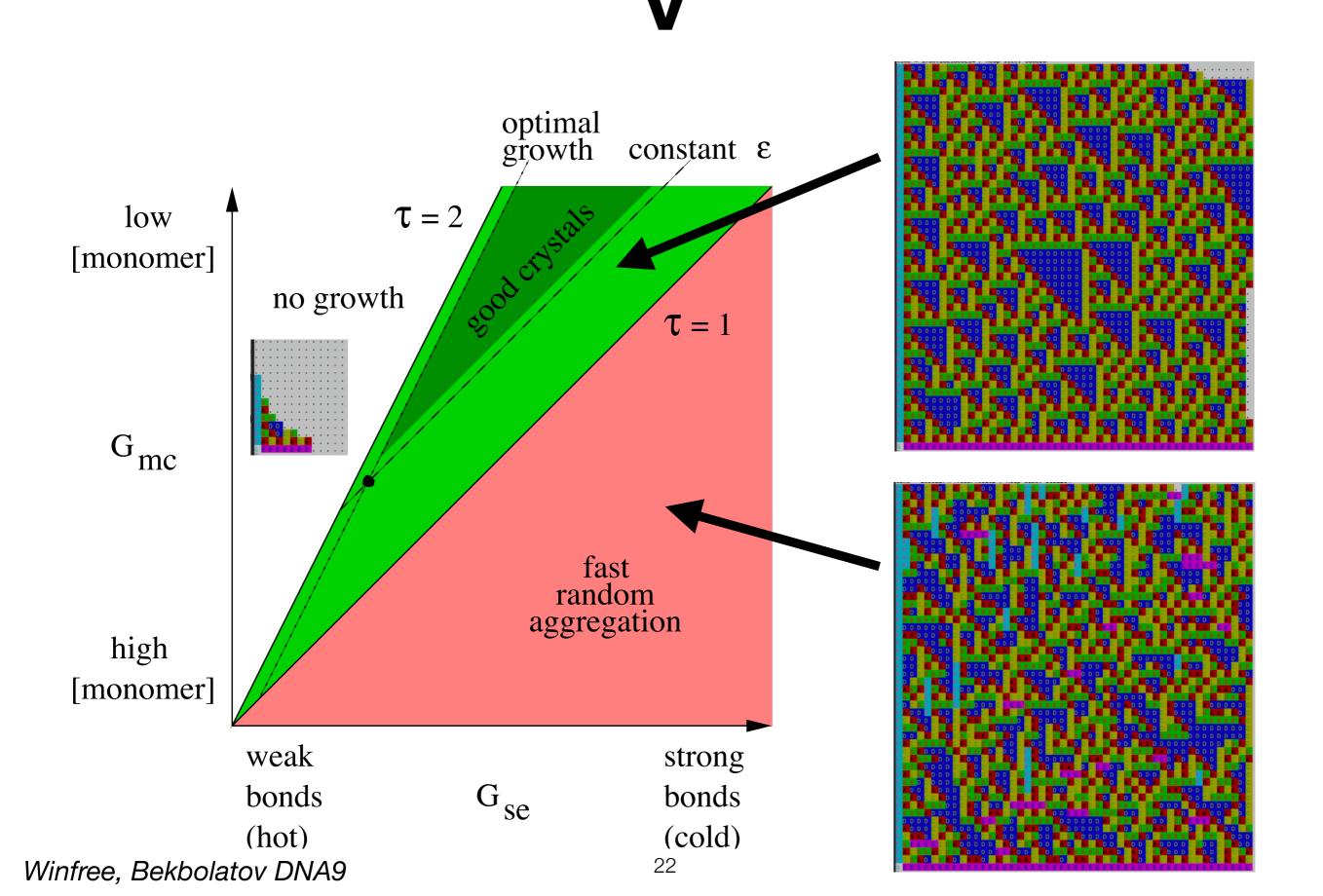
$$k_f \cdot [\ Strand\ ]$$

$=$

$$k_f \cdot \mathbf{e}^{-Gmc}$$

(mainly entropy)

Detachment rate

$=$

$$k_f \cdot \mathbf{e}^{-(b \cdot Gse)}$$
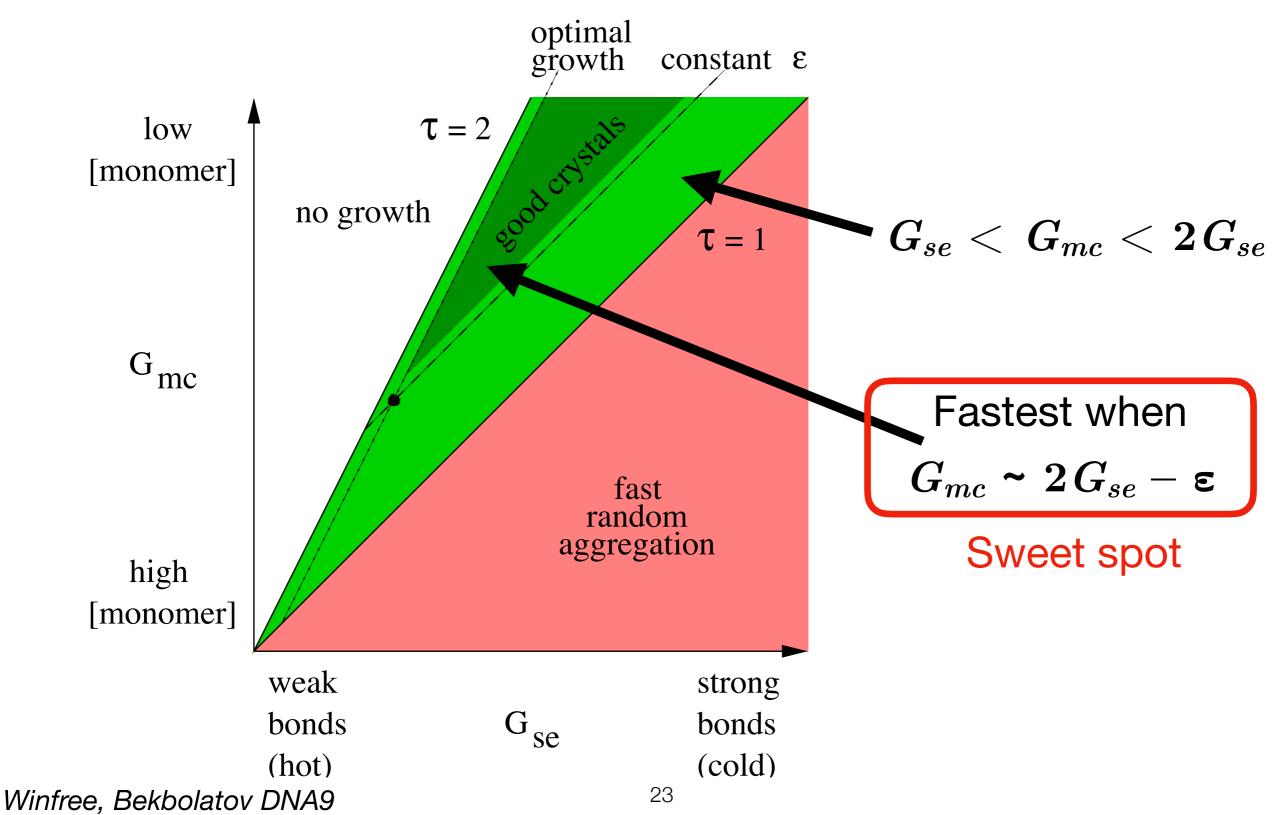
where b is the number of bonds
and $G_{se} = \Delta G / RT$
the bonding unit energy in RT units
(mix of entropy and enthalpy)

$mc = $ **m**onomer **c**oncentration
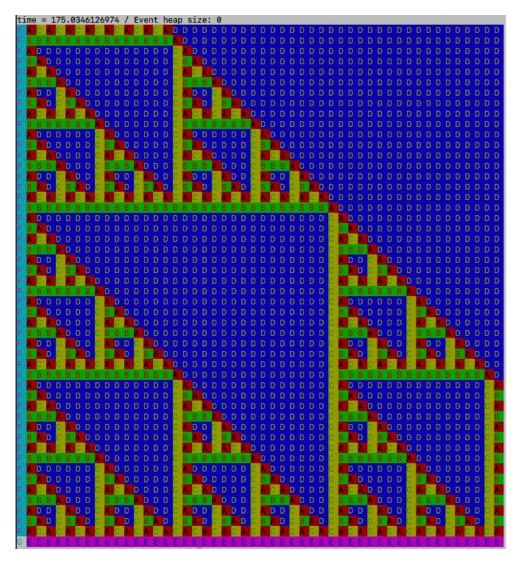
$se = $ **s**ticky **e**nd bond strength

Winfree, Bekbolatov DNA9

21

V

low
[monomer]

optimal
growth

constant ε

$\tau = 2$

no growth

good crystals

$\tau = 1$

G_{mc}

fast
random
aggregation

high
[monomer]

weak
bonds
(hot)

G_{se}

strong
bonds
(cold)

Winfree, Bekbolatov DNA9

Simulations

Winfree, Bekbolatov DNA9
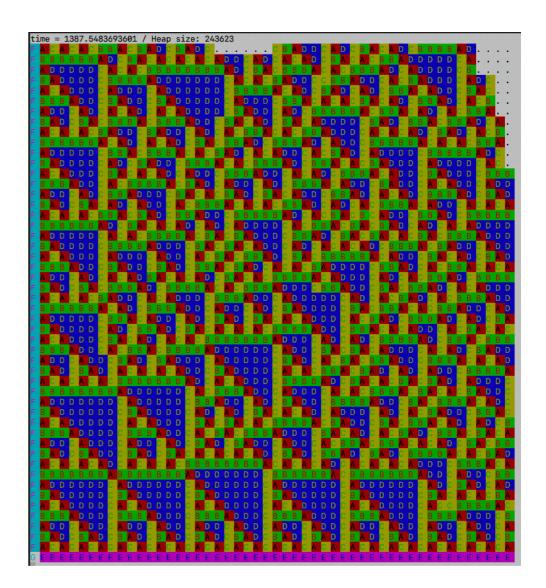
Minimzing errors

Desired

Obtained

24

Proofreading tiles

(a)

tile X $\implies$ 2x2 block X (4 tiles)

rule tiles | boundary tiles

(b) 0+0=0 0+1=1 1+0=1 1+1=0

(c)

(d)
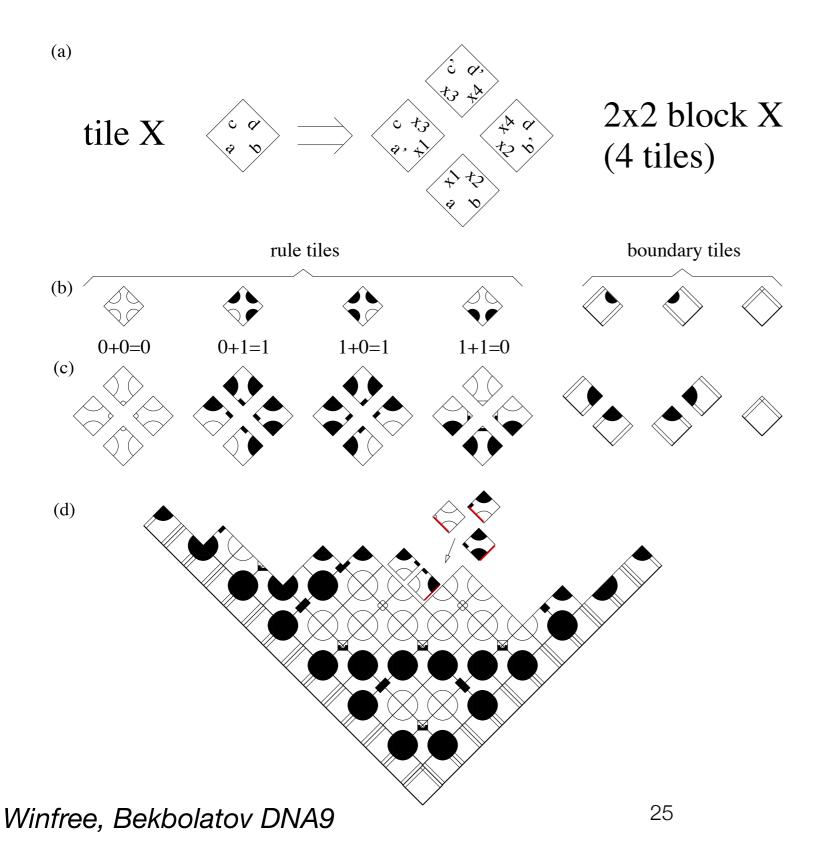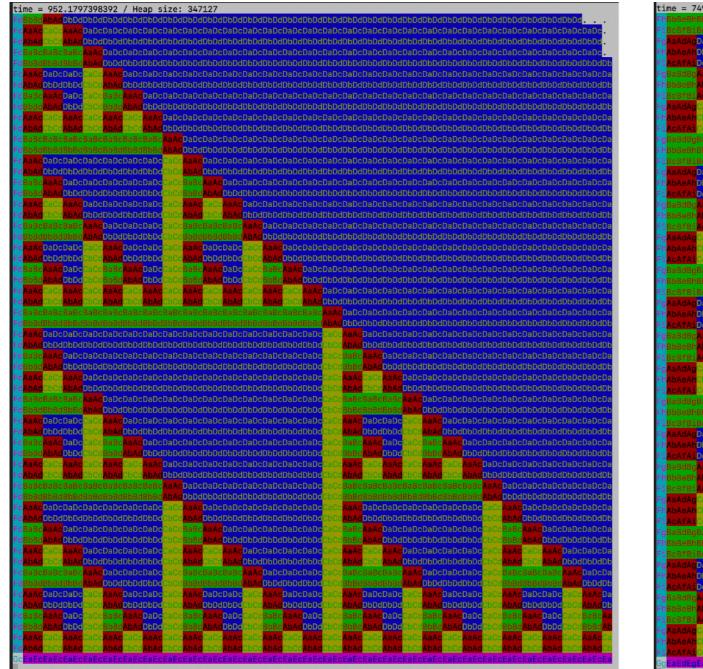
- Cut every tile into k x k tiles

- Now, you need to **make an other error** to compensate for an error

- The error rate is squared for $k = 2$!

Winfree, Bekbolatov DNA9

Proofreading tiles

$k = 2$

$k = 3$

Proofreading tiles compared to other tiles

(a) DX motif

(b) TX motif

(c) SST

(d) SST proofreading

Implementing boolean circuits
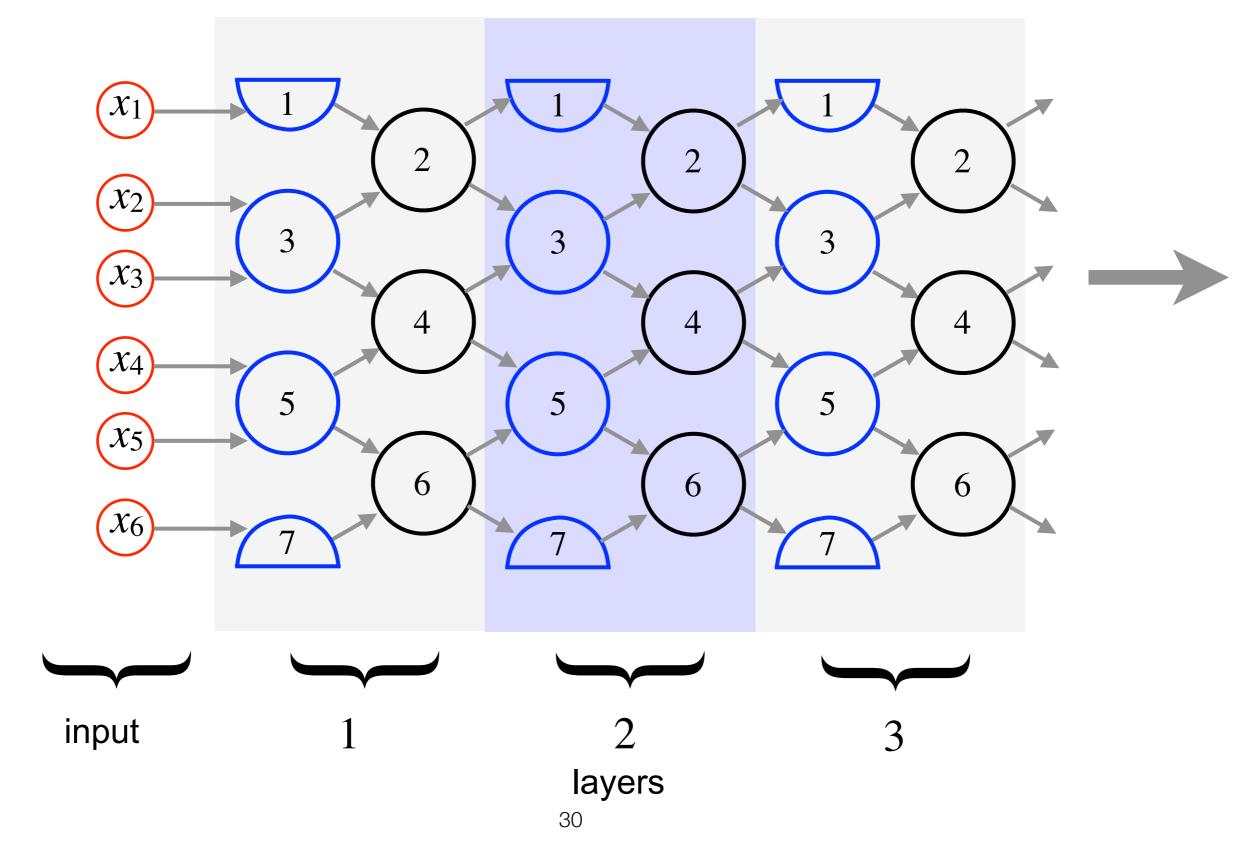
Tile as gates

4 domains = 4 glues

i_1 = **b9**   **a9** = o_1

i_2 = **a8***   **b8*** = o_2

i_1 o_1

r

i_2 o_2

gate g_r
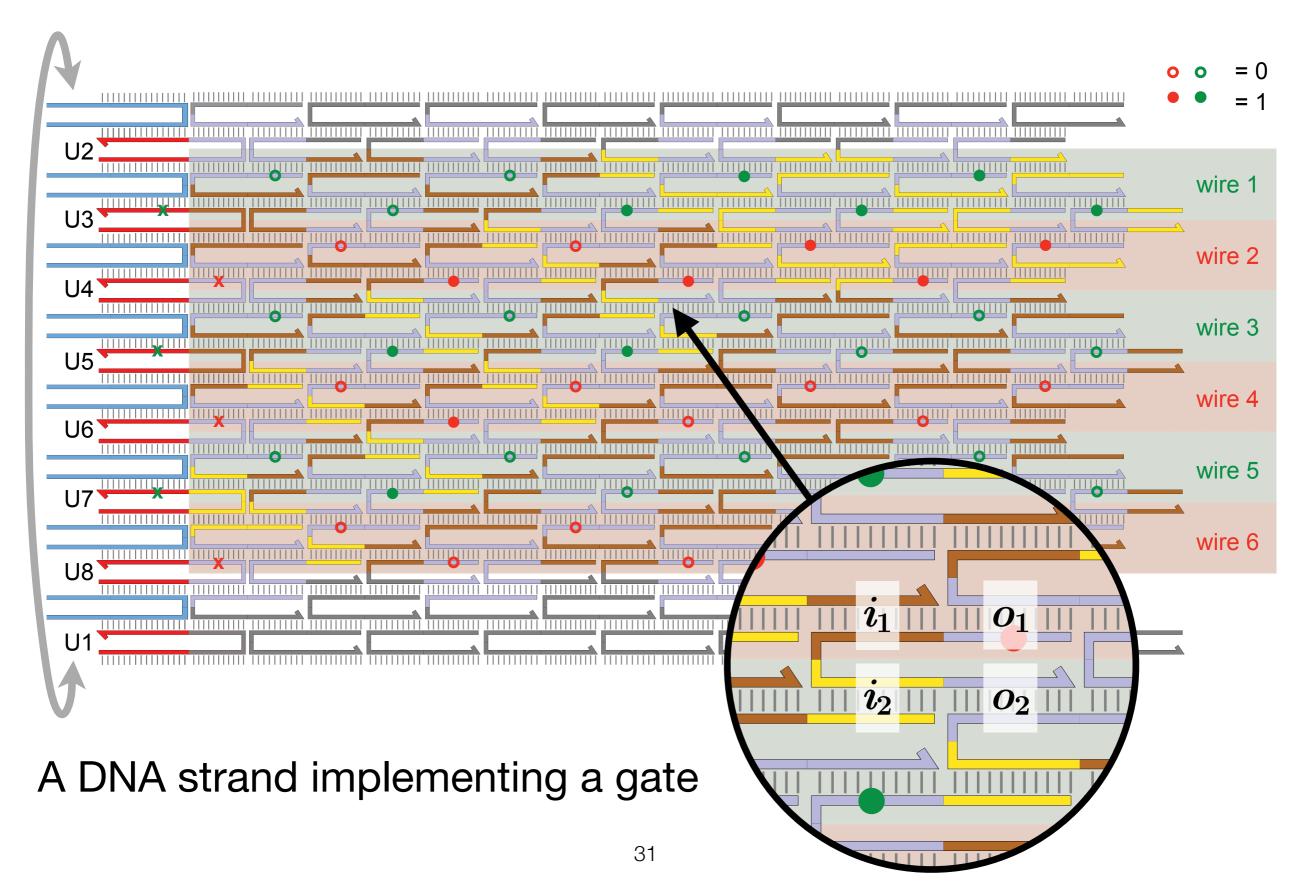
Tiles assembly is a rewriting system
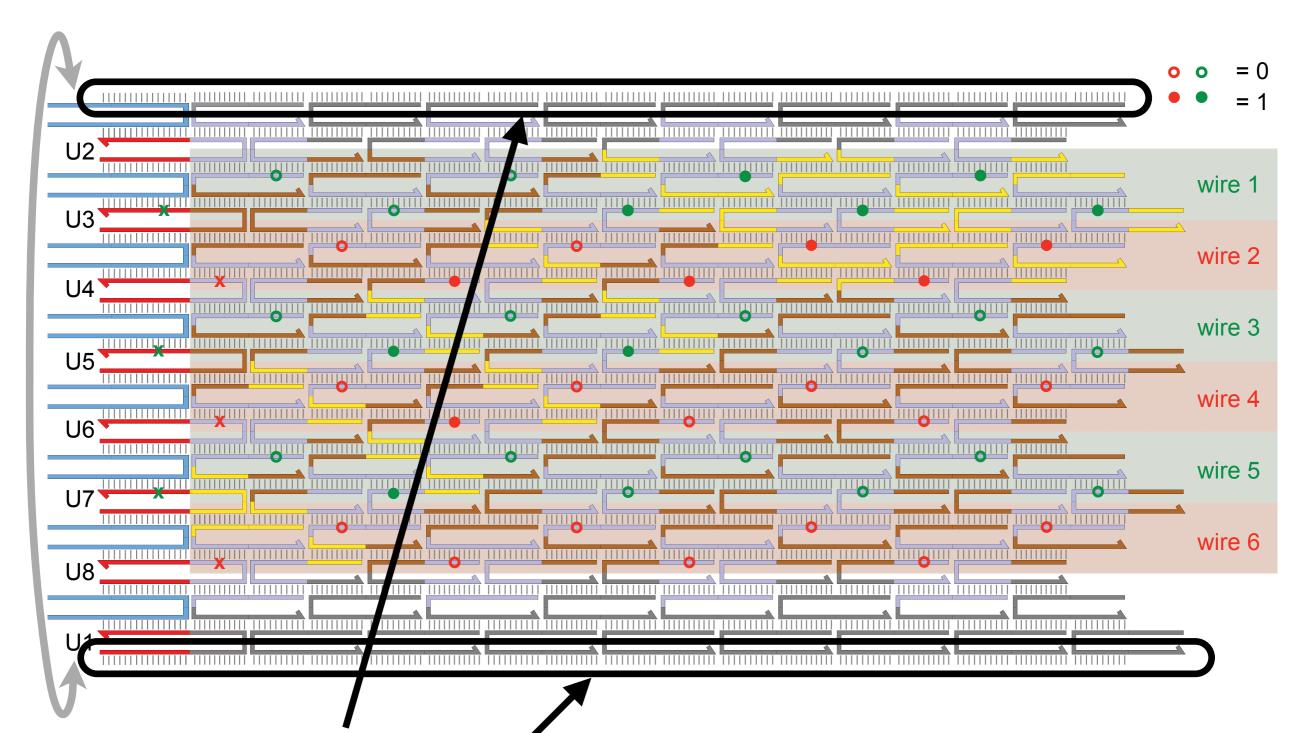
DNA nanotube circuit model

DNA nanotube circuit model

A DNA strand implementing a gate
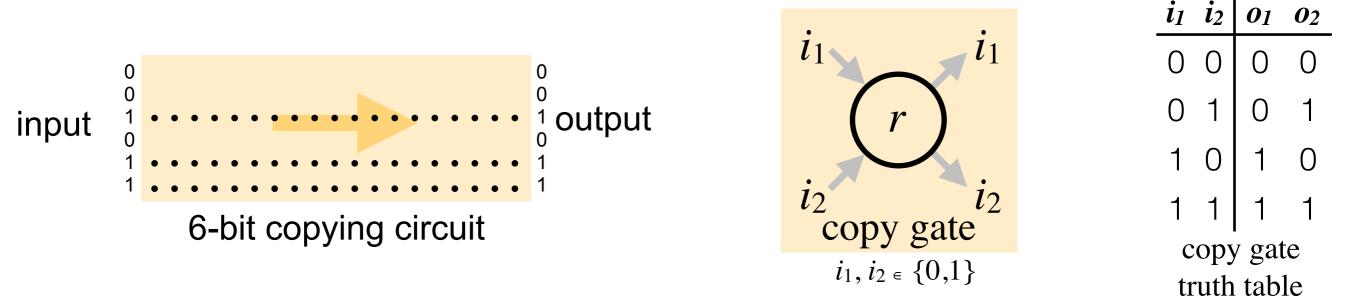
DNA nanotube circuit model
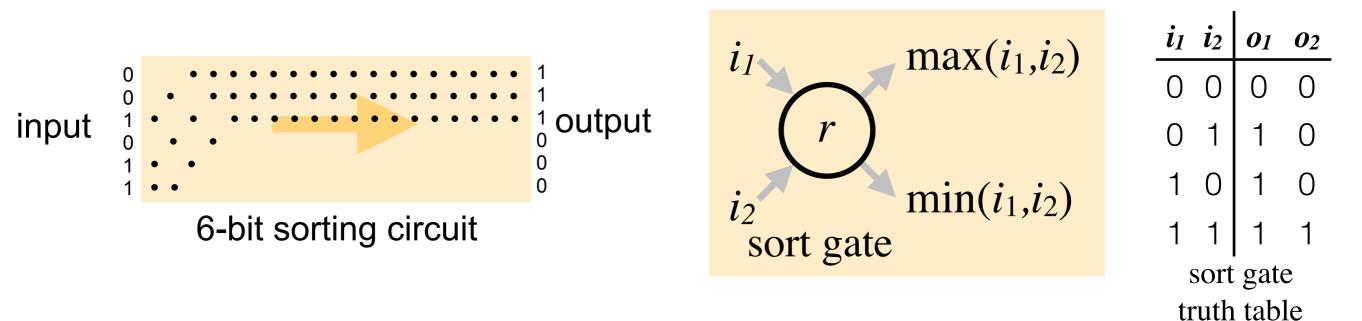
The *seam* which can be *unzipped* to flatten the assembly for imaging

Example nanotube circuits

- **n-bit copying**: $n+1$ copy gates

input   0
         0
         1
         0
         1
         1

output   0
         0
         1
         0
         1
         1

6-bit copying circuit

copy gate

$i_1, i_2 \in \{0,1\}$

i_1	i_2	o_1	o_2
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	1

copy gate
truth table
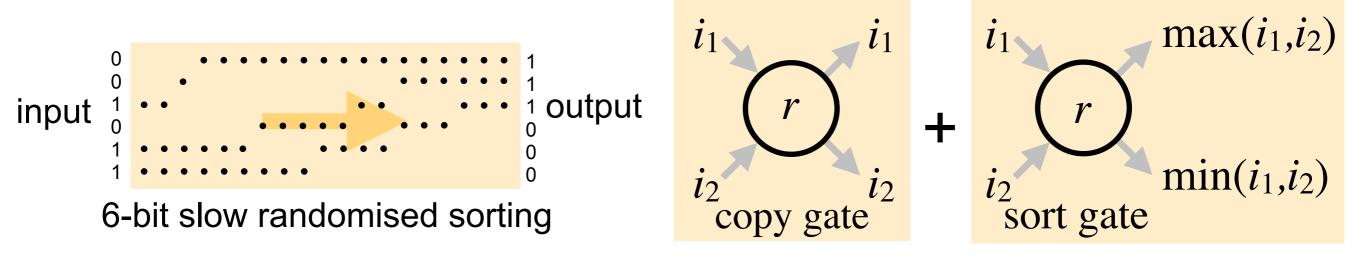
- **n-bit binary sorting:** $n+1$ sort gates

input   0
         0
         1
         0
         1
         1

output   1
         1
         1
         0
         0
         0

6-bit sorting circuit

sort gate

$\max(i_1, i_2)$

$\min(i_1, i_2)$

i_1	i_2	o_1	o_2
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

sort gate
truth table

Note that 2 gates are single input, single output
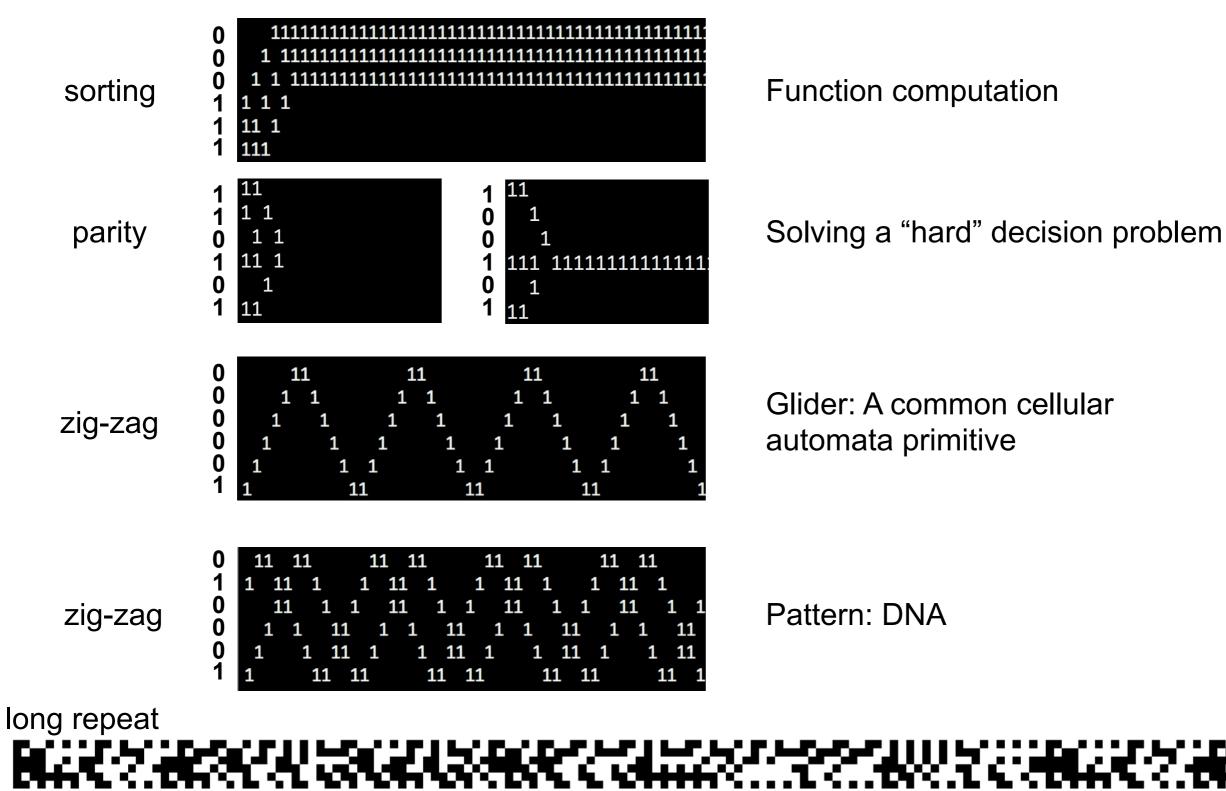
Example nanotube circuits

- **Lazy sorting**! Take the union of the copy gate set and the sort gate set. Copying fights to slow down the sorting process, but assuming a fair execution, sorting will eventually win.

6-bit slow randomised sorting

copy gate + sort gate

- Since, in any given circuit, each gate "knows" its row number r, we will also write circuits (programs) that exploit this feature, do something that is interesting *and* (more importantly) provably impossible without that feature

Circuits

sorting — Function computation

parity — Solving a "hard" decision problem

zig-zag — Glider: A common cellular automata primitive

zig-zag — Pattern: DNA

long repeat

Rule 110
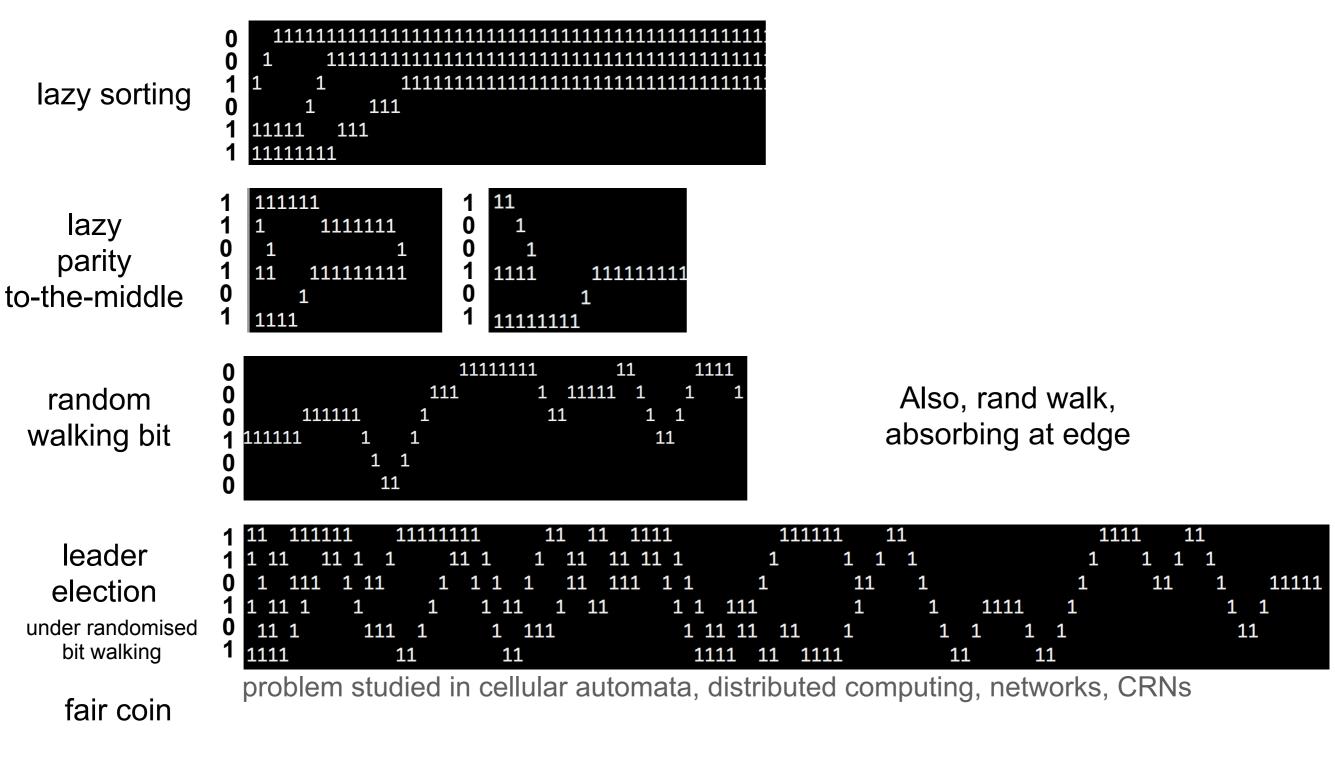
Behaviour: 63 layers to see the same thing twice!

Damien Woods

35

Circuits: randomised

lazy sorting

```
0  11111111111111111111111111111111111111111111111
0   1         1111111111111111111111111111111111111
1  1      1        1111111111111111111111111111111111
0      1       111
1  11111    111
1  1111111
```

lazy parity to-the-middle

```
1  111111                    1  11
1  1      1111111             0   1
0   1          1             0    1
1  11    11111111            1  1111      111111111
0      1                     0        1
1  1111                      1  11111111
```

random walking bit

```
0                 11111111         11        1111
0              111          1  11111  1    1    1
0      111111         1            11      1  1
1 111111       1    1                      11
1               1  1
0               11
0
```

Also, rand walk,
absorbing at edge

leader election

under randomised
bit walking

```
1  11  111111    11111111      11  11  1111        111111    11           1111    11
1  1 11   11 1  1     11 1   1  11  11 11 1        1      1 1 1          1     1   1 1
0   1  111  1 11    1  1 1  1    11  111  1 1      1       11  1        1       11    1    11111
1  1 11 1    1      1    1 11  1  11      1 1  111      1      1  1111    1      1 1
0   11 1     111 1      1  111         1 11 11  11    1       1  1    1 1            11
1  1111        11       11        1111  11 1111      11       11
```

problem studied in cellular automata, distributed computing, networks, CRNs

fair coin

Randomised programs may be a useful tool to calculate energetics of tile binding, or groups of tiles binding, from AFM data

A nice method to assess the quality of our sequence design

Circuits

zig-zag

Glider: A common cellular automata primitive

Pattern: Monotone / horizontally connected

Nonmontonic widely-spaced patterns are provably impossible in the deterministic circuit model

Diamonds are forever
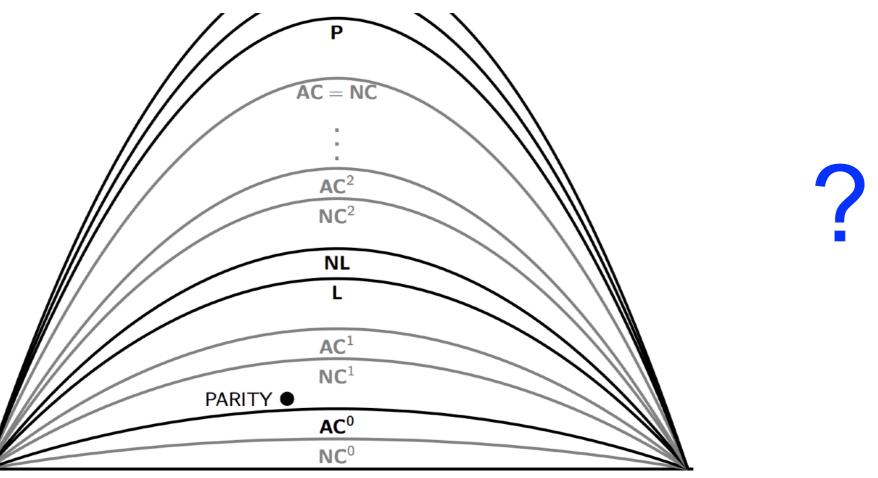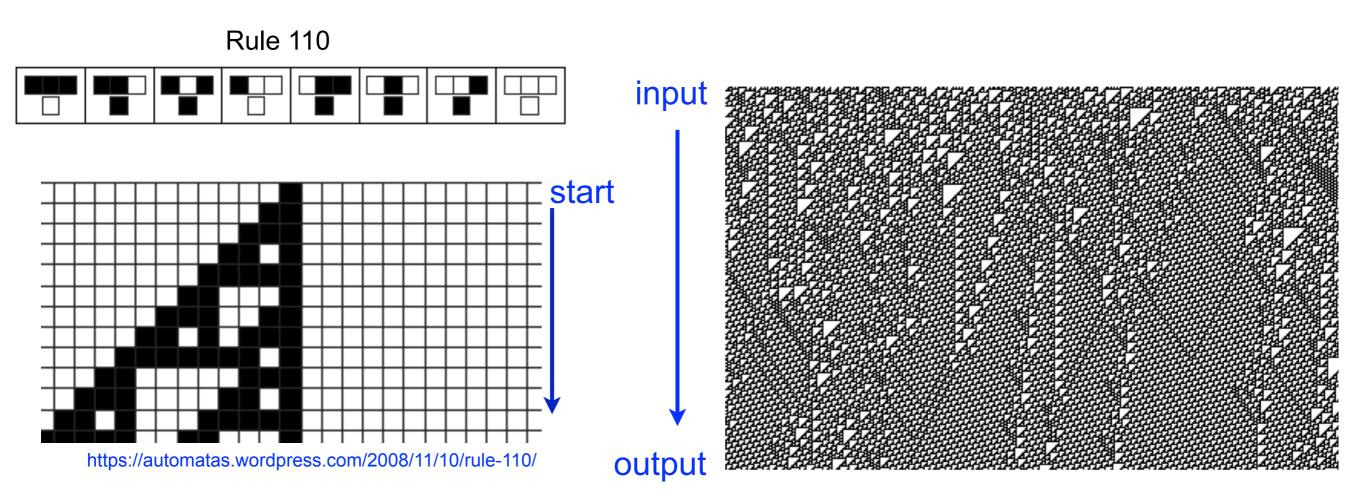
Blowing bubbles

Computational power of DNA
(DNA = DNA nanotube algorithms)

- What is the computational power of our circuit model?

- With n input bits, depth-2 layer, and poly(n) depth circuit, what can be solved?

  - No more than P (proof: simulate poly(n) depth circuit in polynomial time on a Turing machine)

  - We've seen already that the model can solve SORTING, PARITY both of which are outside AC^0

$?$

Rule 110

Rule 110

https://automatas.wordpress.com/2008/11/10/rule-110/

input

start

output

- **Theorem:** Rule 110 is an **efficient** and **general purpose** computer

Neary, Woods.   Cook. Complex
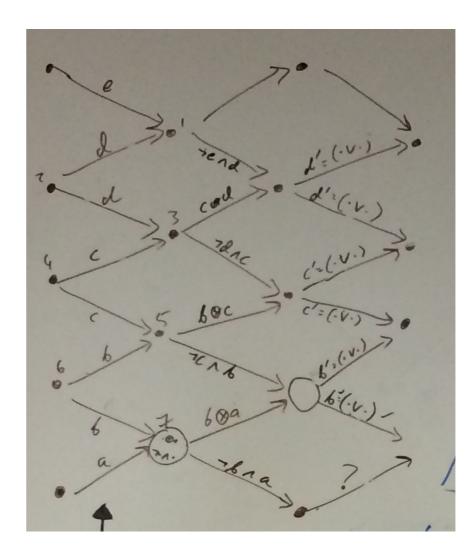ICALP 2006   Systems. 15:1-40 2004

Computational power of DNA
(DNA = DNA nanotube algorithms)

- What is the computational power of our circuit model?

- With n input bits, depth-2 layer, and poly(n) depth circuit, what can be solved?

  - No more than P. Proof: simulate poly(n) depth circuit in polynomial time on a Turing machine

  - All of P: Proof: simulate Rule 110

$$\begin{array}{ll} \text{c b a} & \text{c b a} \\ F(0,0,0) = 0 & F(1,0,0) = 0 \\ F(0,0,1) = 1 & F(1,0,1) = 1 \\ F(0,1,0) = 1 & F(1,1,0) = 1 \\ F(0,1,1) = 1 & F(1,1,1) = 0 \end{array}$$

Computational power of DNA
(DNA = DNA nanotube algorithms)

- What is the computational power of our circuit model?

- With n input bits, depth-2 layer, and poly(n) depth circuit, what can be solved?

  - Answer: Exactly P, via Rule 110 simulation

T. Neary, D. Woods. P-completeness of cellular automaton Rule 110. ICALP 2006. Springer LNCS 4051(1):132-143
Cook, M.: Universality in elementary cellular automata. Complex Systems 15 (2004) 1–40