UNIVERSITY OF OXFORD

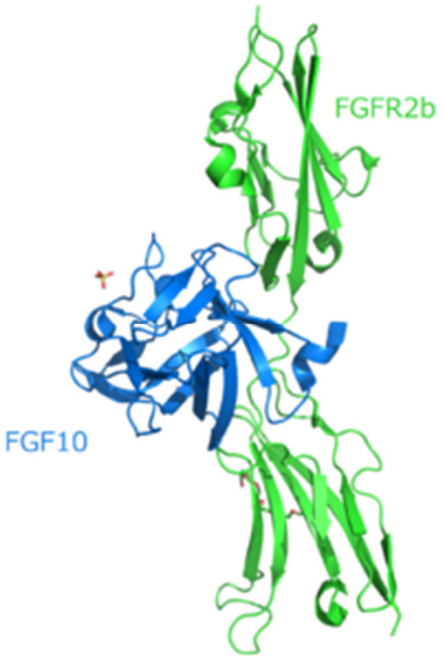
# Successes and challenges of modelling and verification at the nanoscale (and some failures too…)

Marta Kwiatkowska, University of Oxford

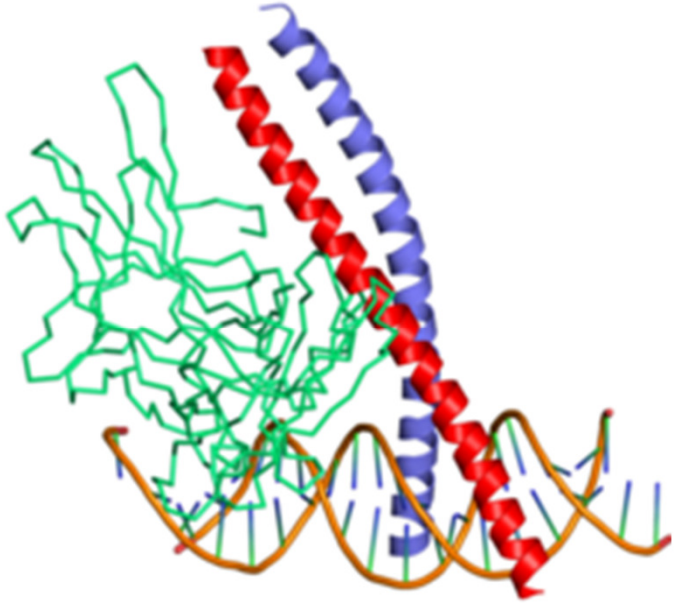
Simons Institute, 13th August 2015

# At the nanoscale…

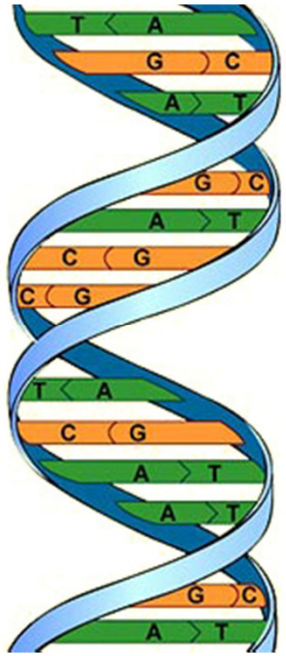
- World of molecules




FGF protein



Fos and Jun bound to DNA

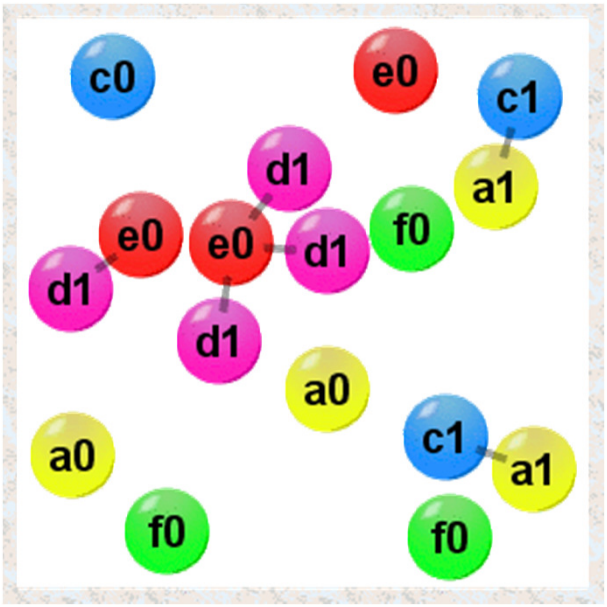


DNA molecule

- Aim to understand their function not only in biological processes, but also as engineering material

# Modelling molecular networks

- Focus on modelling dynamics and analysis of behaviours
  - networks of molecules
  - molecular interaction
  - molecular motion
  - self-assembly




- Rather than
  - geometry
  - structure
  - sequence

- Chemical reaction networks

- Emphasis on quantitative/probabilistic characteristics
- Stochasticity essential for low molecular counts

# Chemical reaction networks

Used to encode a real or hypothetical mechanism

1: FGF binds/releases FGFR

FGFR + FGF $\rightarrow$ FGFR:FGF $\quad k_1 = 5e+8\ M^{-1}s^{-1}$

FGFR + FGF $\leftarrow$ FGFR:FGF $\quad k_2 = 0.002\ s^{-1}$

2: Relocation of FGFR (whilst phosphorylated)

FGFR $\rightarrow$ $\quad k_3 = 0.1\ s^{-1}$

Can map to different semantics/representation

# Chemical reaction networks

Used to encode a real or hypothetical mechanism

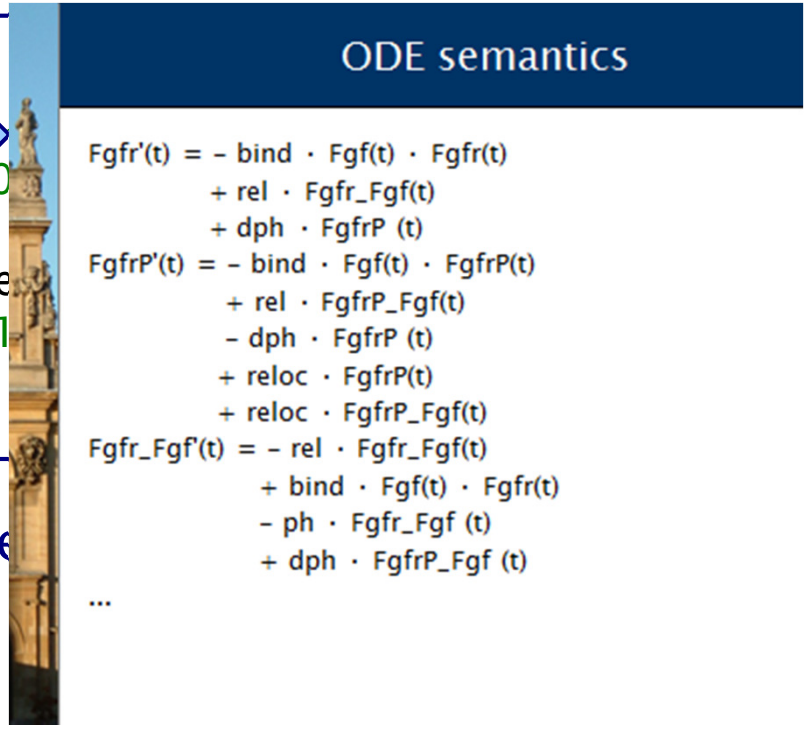
1: FGF binds/releases FGFR

FGFR + FGF → FGFR:FG

FGFR + FGF ← FGFR:FGF $k_2$=0.0

2: Relocation of FGFR (whilst phosphorylate

FGFR → $k_3$=0.1



ODE semantics

$$
\begin{aligned}
Fgfr'(t) &= -bind \cdot Fgf(t) \cdot Fgfr(t) \\
&\quad + rel \cdot Fgfr\_Fgf(t) \\
&\quad + dph \cdot FgfrP(t) \\
FgfrP'(t) &= -bind \cdot Fgf(t) \cdot FgfrP(t) \\
&\quad + rel \cdot FgfrP\_Fgf(t) \\
&\quad - dph \cdot FgfrP(t) \\
&\quad + reloc \cdot FgfrP(t) \\
&\quad + reloc \cdot FgfrP\_Fgf(t) \\
Fgfr\_Fgf'(t) &= -rel \cdot Fgfr\_Fgf(t) \\
&\quad + bind \cdot Fgf(t) \cdot Fgfr(t) \\
&\quad - ph \cdot Fgfr\_Fgf(t) \\
&\quad + dph \cdot FgfrP\_Fgf(t) \\
&\ldots
\end{aligned}
$$

# Chemical reaction networks

Used to encode a real or hypothetical mechanism

1: FGF binds/releases FGFR

FGFR + FGF → FGFR:FG

FGFR + FGF ← FGFR:FGF $k_2=0.0$

2: Relocation of FGFR (whilst phosph

FGFR →

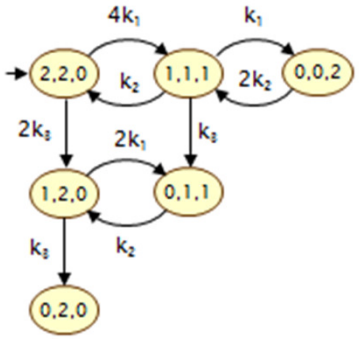
Can map to different semantics/

## ODE semantics

Fgfr'(t) = – bind · Fgf(t) · Fgfr(t)
+ rel · Fgfr_Fgf(t)

## CTMC semantics

- CTMC with state space
  - (|FGFR|,|FGF|,|FGFR:FGF|)
  - initially (2,2,0)




- under assumption of mass action kinetics

8

# Chemical reaction networks

Used to encode a real or hypothetical mechanism

1: FGF binds/releases FGFR

FGFR + FGF → FGFR:FG

FGFR + FGF ← FGFR:FGF $k_2=0.0$

2: Relocation of FGFR (whilst phosph

FGFR →

Can map to different

**ODE semantics**

Fgfr'(t) = – bind · Fgf(t) · Fgfr(t)
+ rel · Fgfr_Fgf(t)

**CTMC semantics**

- CTMC with state space
  - (|FGFR|,|FGF|,|FGFR:FGF|)
  - initially (2,2,0)

**PRISM reactive modules**

```
module fgfr

  fgfr  : [0..1] init 0; // 0 - free, 1 - bound
  phos  : [0..1] init 0; // 0 - unphosphorylated, 1 - phosphorylated
  reloc : [0..1] init 0; // 0 - not relocated, 1- relocated

  [bnd] reloc=0 ∧ fgfr=0 → k1 : (fgfr'=1); // FGF and FGFR bind
  [rel]  reloc=0 ∧ fgfr=1 → k2 : (fgfr'=0); // FGF and FGFR release
  [] reloc=0 ∧ fgfr =1 ∧ phos =0 → k3 : (phos'=1); // FGFR phosphor.
  [] reloc=0 ∧ phos=1     → k4 : (phos'=0); // FGFR dephosphorylates
  [] reloc=0 ∧ phos=1     → k5 : (reloc'=1); // FGFR relocates

endmodule
```

8

# Chemical reaction networks

Used to encode a real or hypothetical mechanism

1: FGF binds/releases FGFR

FGFR + FGF → FGFR:FG

FGFR + FGF ← FGFR:FGF $k_2=0.0$

2: Relocation of FGFR (whilst phosph

FGFR →

Can map to different

ODE semantics

Fgfr'(t) = – bind · Fgf(t) · Fgfr(t)
+ rel · Fgfr_Fgf(t)

CTMC semantics

- CTMC with state space
  - (|FGFR|,|FGF|,|FGFR:FGF|)
  - initially (2,2,0)

PRISM reactive modules

module fgfr

SBML code

```
<listOfSpecies>
  <species id="FGFR" initialConcentration="1" ... />
  <species id="FGF" initialConcentration="1" ... /> ...
</listOfSpecies>
<reaction id="Reaction1" reversible="true">
  <listOfReactants>
    <speciesReference species="FGFR" />...
  </listOfReactants>
  <listOfProducts>
```

# Chemical reaction networks

Used to encode a real or hypothetical mechanism

1: FGF binds/releases FGFR

FGFR + FGF → FGFR:FGF $k_1$=5e+8 $M^{-1}s^{-1}$

FGFR + FGF ← FGFR:FGF $k_2$=0.002 $s^{-1}$

2: Relocation of FGFR (whilst phosphorylated)

FGFR → $k_3$=0.1 $s^{-1}$

Can map to different semantics/representation

- Now can apply probabilistic model checking to obtain model predictions…
  - software tools exist and are well used, e.g. PRISM
- Sounds easy?

# The PRISM model checker

- Inputs CTMC models in reactive modules or SBML
- and specifications given in probabilistic temporal logic CSL
  - what is the probability that the concentration reaches min?

    $P_{=?}$ [F c≥min]
  - in the long run, what is the probability that the concentration remains stable between min and max?

    $S_{=?}$ [(c ≥min)∧(c≤max)]

- Then computes model predictions via
  - exhaustive analysis to compute probability and expectations over time (with numerical precision)
  - or probability estimation based on simulation (approximate, with confidence interval)
- See www.prismmodelchecker.org

PRISM 4.0:Verification of Probabilistic Real-time Systems, Kwiatkowska *et al*, In*Proc.CAV'11*

# What's involved

- Modelling formalisms
  - chemical reaction networks, continuous-time Markov chains, reactive modules, stochastic Petri nets, pi-calculus…
- Specification notations
  - temporal logic (LTL, CTL, PCTL, CSL)
- Analysis methods
  - model construction/extraction/reduction, graph-theoretical algorithms, symbolic (BDD/MTBDD), symbolic (SAT/SMT), linear equation solving, uniformisation, fast adaptive uniformisation, LNA, ODE solving, stochastic simulation, model checking, probabilistic model checking, statistical model checking, parallelisation…
- Distinctive CS influence
  - abstractions, logic, general purpose formalisms and languages, symbolic algorithms and representations…

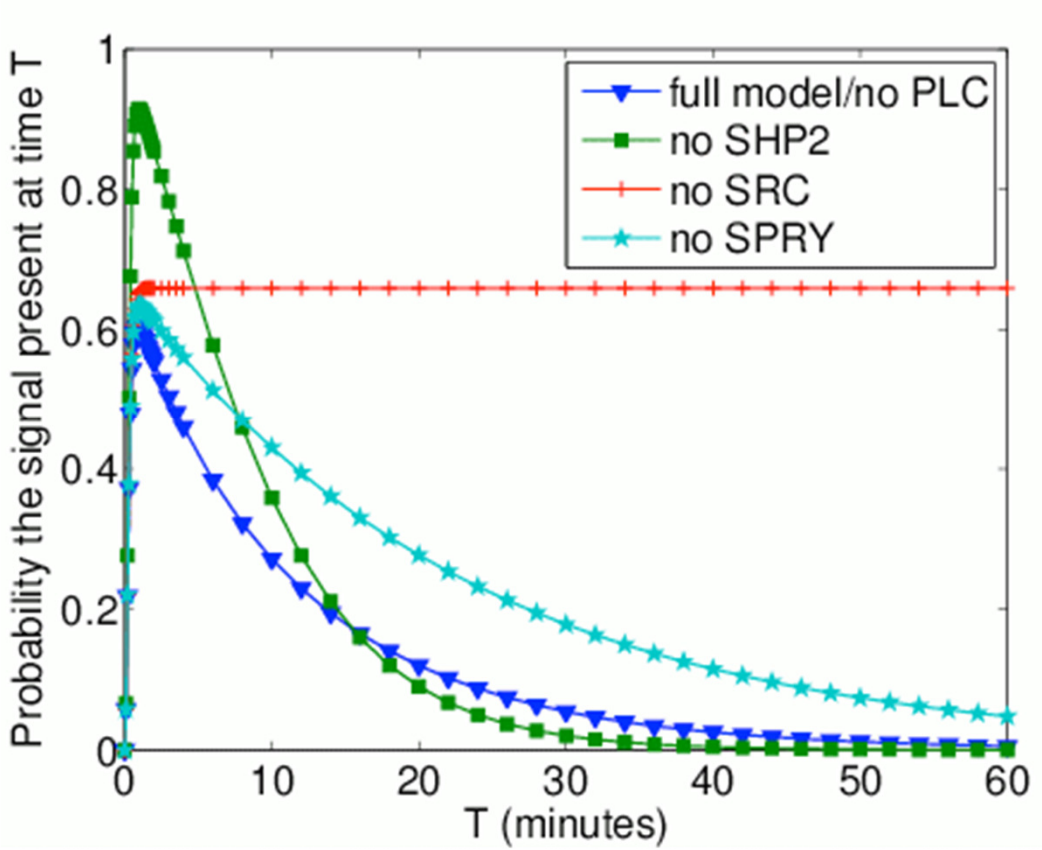
# Case study 1: FGF pathway

- Fibroblast Growth Factor (FGF) pathway
  - regulator of skeletal development
- Biological challenges
  - unknown function of molecules
  - expensive experimental scenarios
- Aim to analyse the dynamics of FGF signalling
  - model different hypothetical regulation mechanisms
  - "in silico genetics"
- Modelling
  - PRISM model highly complex, 2m states (one molecule each)
  - ODE model > 300 equations, need simplifications
- Predictions
  - new, experimentally validated [Sandilands et al, 2007]

Probabilistic Model Checking of Complex Biological Pathways, Heath *et al*, Theoretical Computer Science 391(3), 2008

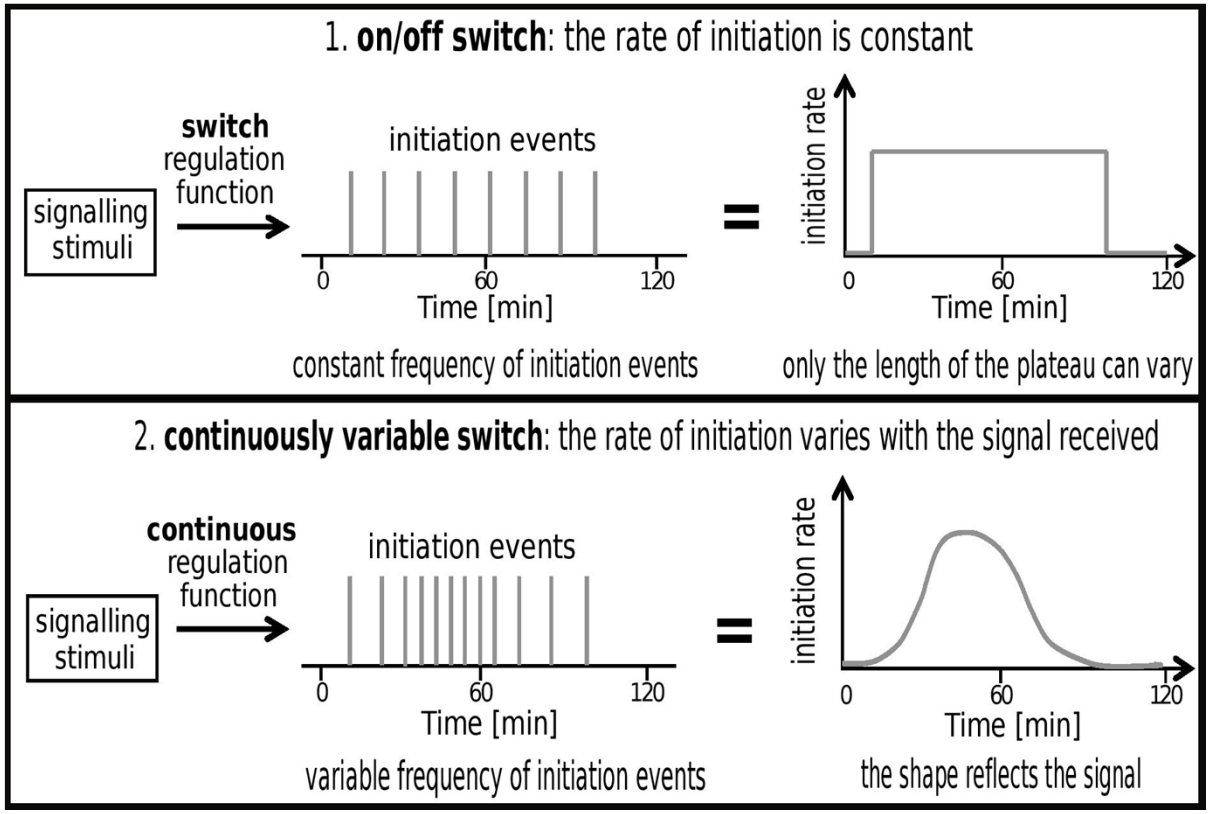
# In silico genetics experiment (FGF)

- SRC prominent determinant of FGF signalling



# Case study 2: Inducible genes

- Immediate early gene induction, e.g. c-fos and c-jun
  - viewed as two-state or continuously variable




Stochastic modelling of the interface between regulatory enzymes and transcriptional initiation at inducible genes, Ceska *et al*, in preparation, 2015

# Inducible genes: results

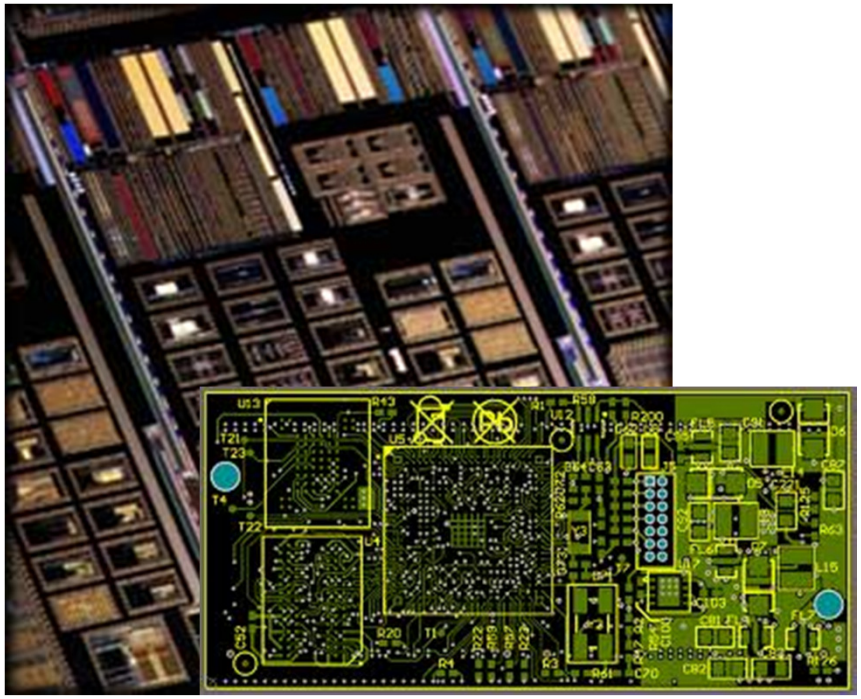
- Modelling approach
  - both types of switch accommodated in the interface (step function and sigmoidal function)
  - fit to experimental activation profiles
- Modelling challenges
  - large population of MAPK signalling vs single copy of gene
  - stochasticity and noise considerations
  - rates determined by kinase activation profile, so inhomogeneous CTMC
  - approximate using piecewise constant CTMCs
- Perform “in silico” comparison of the two switches
- Obtain reasonable predictions that support the hypothesis
  - continuous switch provides a more viable controlling mechanism for IE genes
  - binary switch fails to reproduce the induction profiles

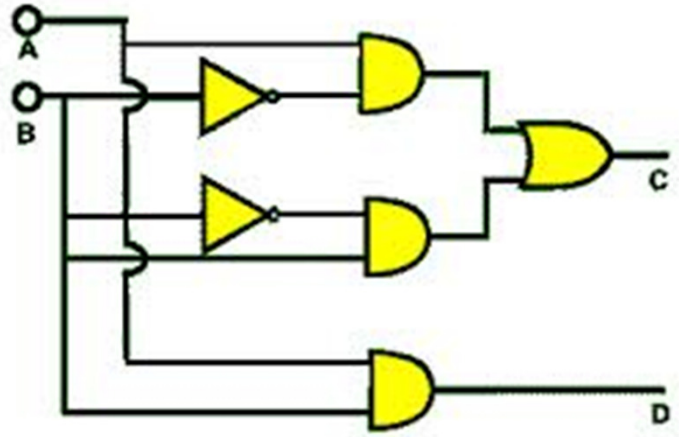
# DNA computation

- DNA: versatile, easily accessible, cheap to synthesise material
- Moore's law, hence need to make devices smaller…
- DNA computation, directly at the molecular level
  - DNA logic circuit designs
  - nanorobotics, via programmable molecular motion
- Many applications envisaged
  - e.g. biosensing, point of care diagnostics, smart therapeutics, …

- Apply quantitative verification and synthesis to
  - automatically find design flaws in DNA computing devices
  - analyse reliability and performance of molecular walkers
  - automatically synthesise reaction rates to guarantee a specified level of reliability
  - develop predictive model of origami folding

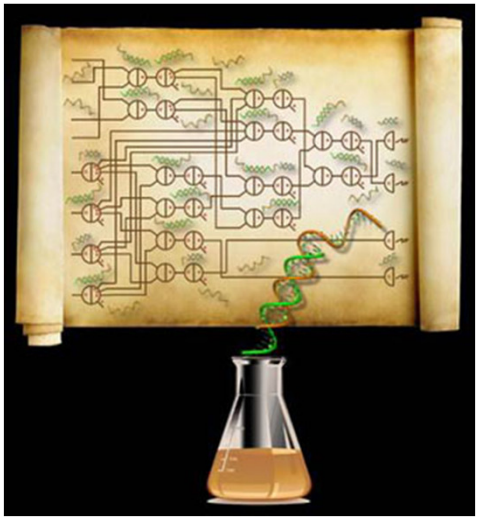
# Digital circuits






- Logic gates realised in silicon
- 0s and 1s are represented as low and high voltage
- Hardware verification indispensable as design methodology

# DNA circuits, in solution



[Qian, Winfree, *Science* 2012]

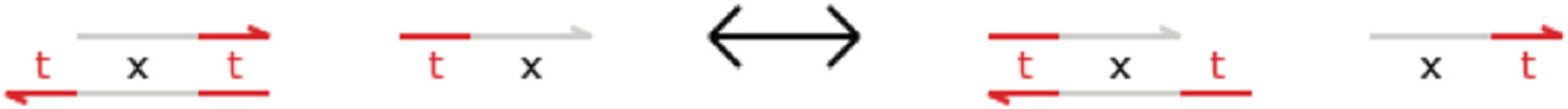
- "Computing with soup" (The Economist 2012)
- Single strands are inputs and outputs
- Circuit of 130 strands computes square root of 4 bit number, rounded down
- 10 hours, but it's a first…

Pop quiz, hotshot: what's the square root of 13?
*Science Photo Library/Alamy*

# Case study 3: DNA transducer gate

- DNA computing with a restricted class of DNA strand displacement structures (process algebra by Cardelli)
  - double strands with nicks (interruptions) in the top strand


  - and two-domain single strands consisting of one toehold domain and one recognition domain

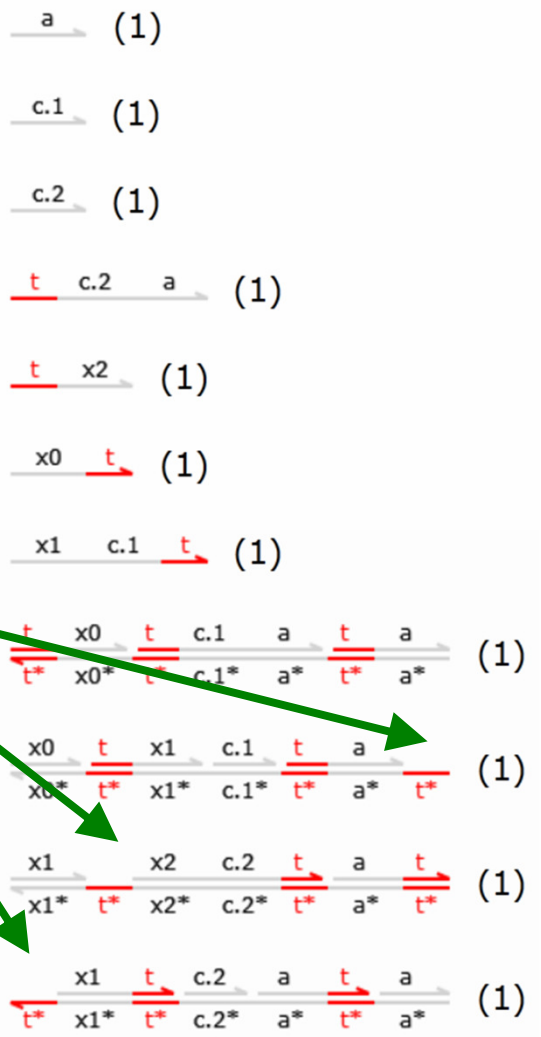



  - "toehold exchange": branch migration of strand <t^ x> leading to displacement of strand <x t^>
- Used to construct transducers, fork/join gates
  - can be formed into cascades
  - all gates in a cascade mixed together…

# DNA transducer flaw

- Cardelli's DNA transducer gate
  - inputs/outputs single strands
  - two transducers connected
- PRISM identifies a bug: 5-step trace to a "bad" deadlock state
  - previously found manually [Cardelli'10]
  - detection now fully automated
- Bug is easily fixed
  - (and verified)

reactive gates



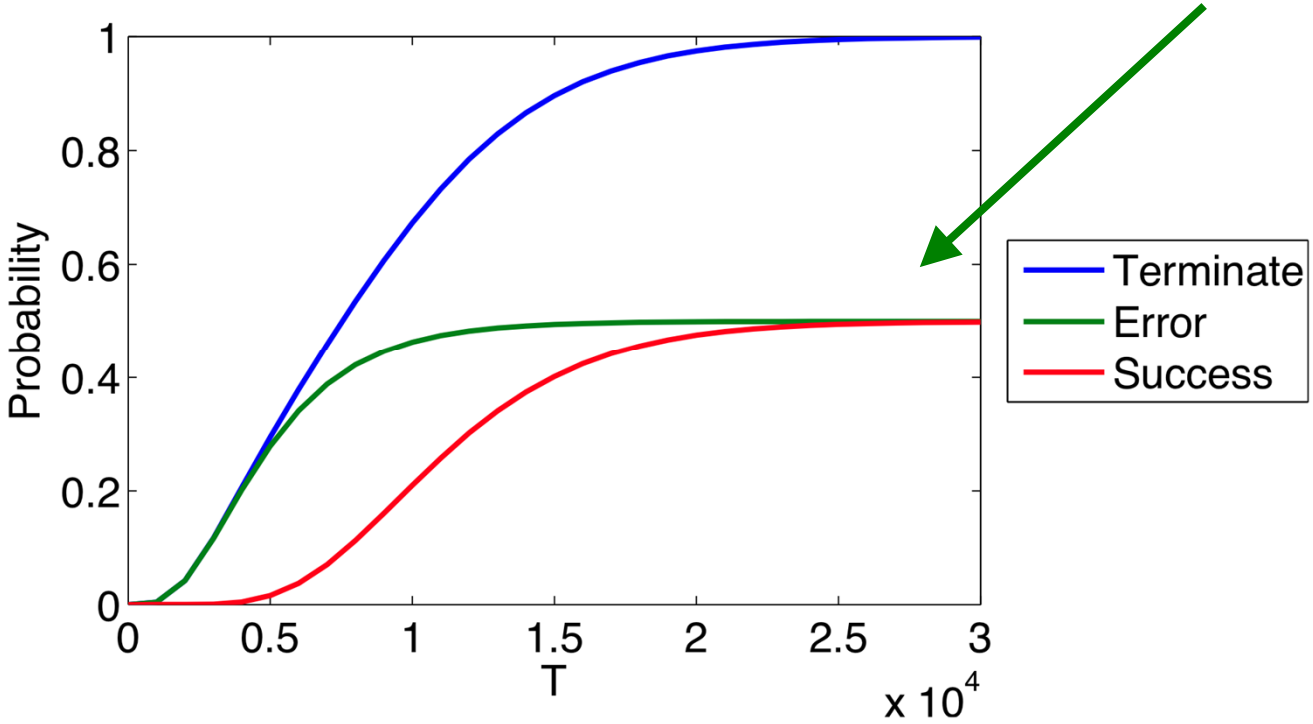

**Counterexample:**
(1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
(0,1,1,0,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
(0,0,1,0,1,1,1,1,1,1,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
(0,0,1,0,1,1,1,1,1,0,0,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
(0,0,1,0,1,1,0,1,0,0,1,1,1,0,0,0,1,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
(0,0,1,0,1,1,0,1,0,0,1,0,1,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)

Design and Analysis of DNA Strand Displacement Devices using Probabilistic Model Checking, Lakin *et al*, Journal of the Royal Society Interface, 9(72), 1470–1485, 2012

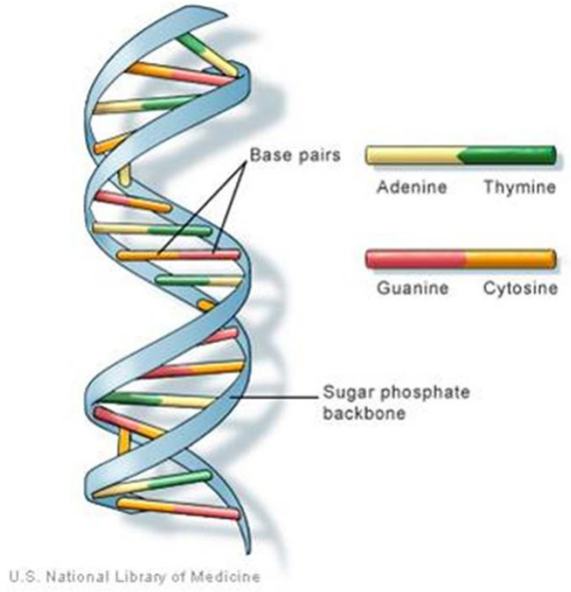
# Quantitative properties

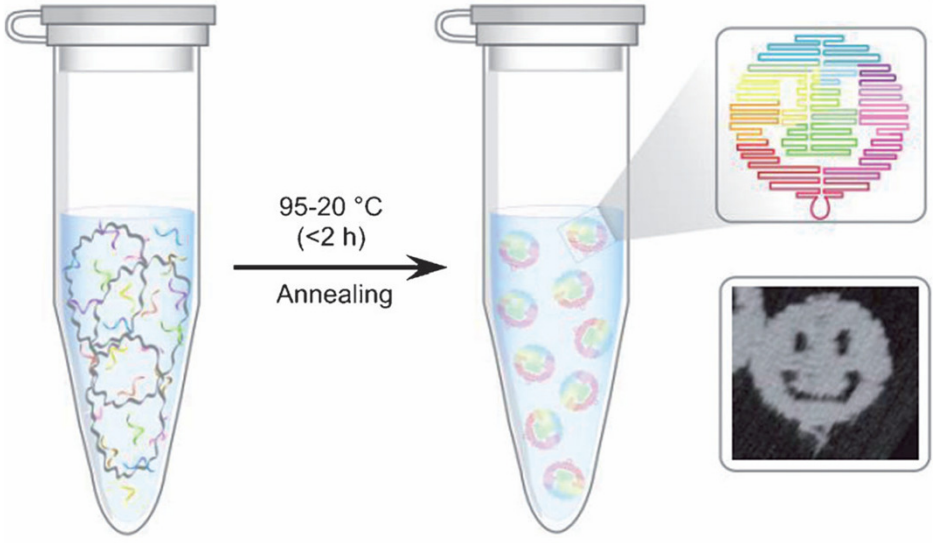
- We can also use PRISM to study the kinetics of the pair of (faulty) transducers:
  - $P_{=?}$ [ $F^{[T,T]}$ "deadlock" ]
  - $P_{=?}$ [ $F^{[T,T]}$ "deadlock" & !"all_done" ]
  - $P_{=?}$ [ $F^{[T,T]}$ "deadlock" & "all_done" ]

success/error equally likely



# DNA nanostructures



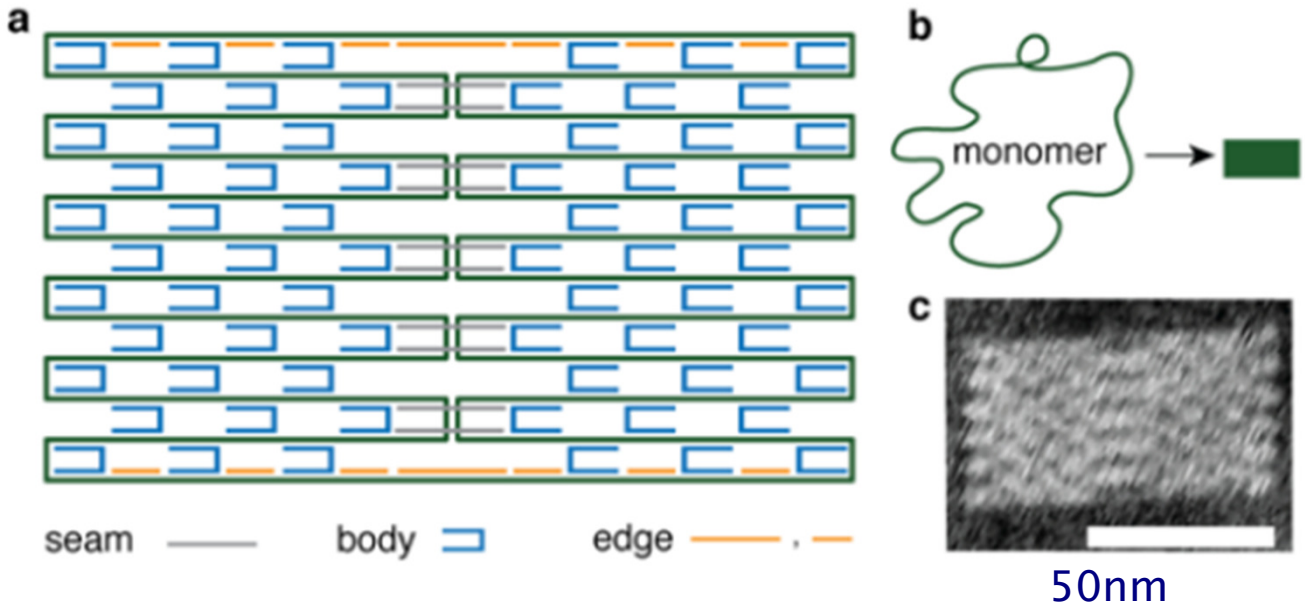




DNA origami

- DNA origami [Rothemund, *Nature* 2006]
  - DNA can self-assemble into structures – "molecular IKEA?"
  - programmable self-assembly (can form tiles, nanotubes, boxes that can open, etc)
  - simple manufacturing process (heating and cooling), not yet well understood

# DNA origami tiles

- DNA origami tiles: molecular breadboard [Turberfield lab]

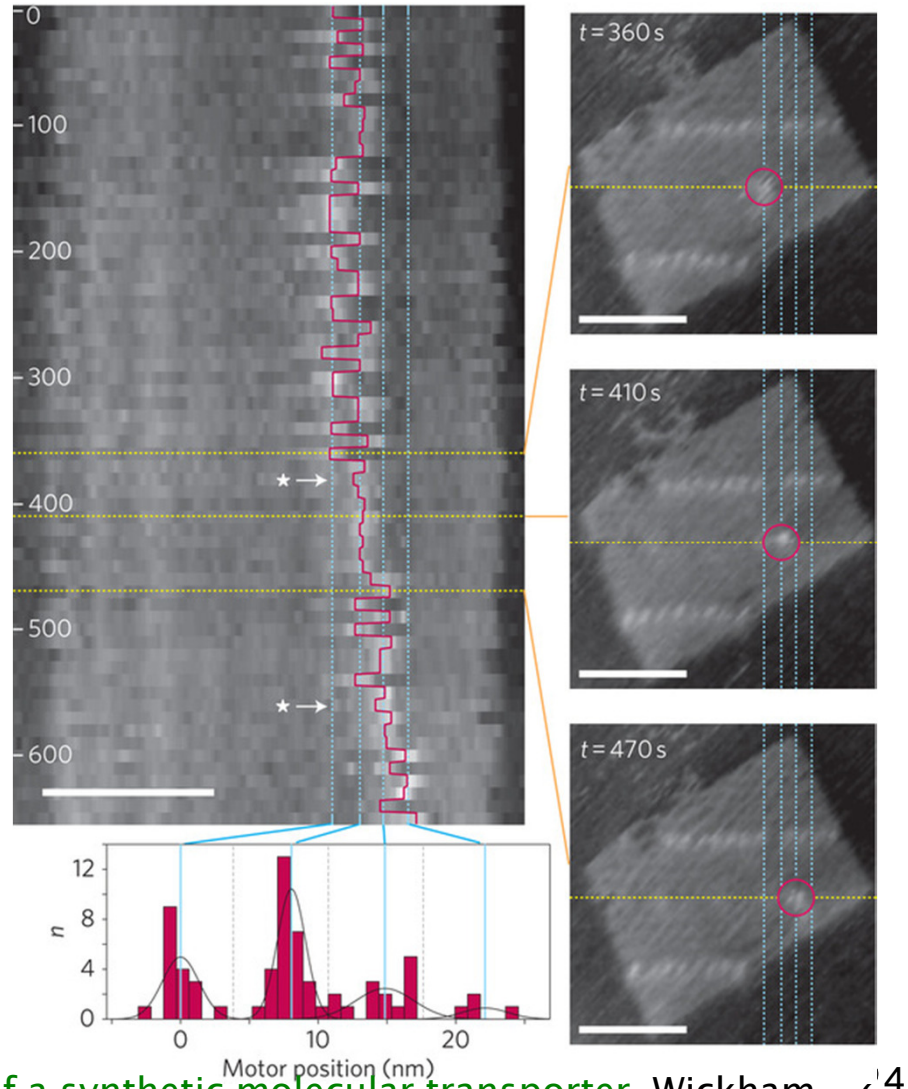



a. Tile design, showing staples 'pinning down' the scaffold and highlighting seam staples
b. Circular single strand (scaffold) that folds into tile
c. AFM image of the tile

Guiding the folding pathway of DNA origami. Dunne, Dannenberg, Ouldridge, Kwiatkowska, Turberfield & Bath, Nature (in press)

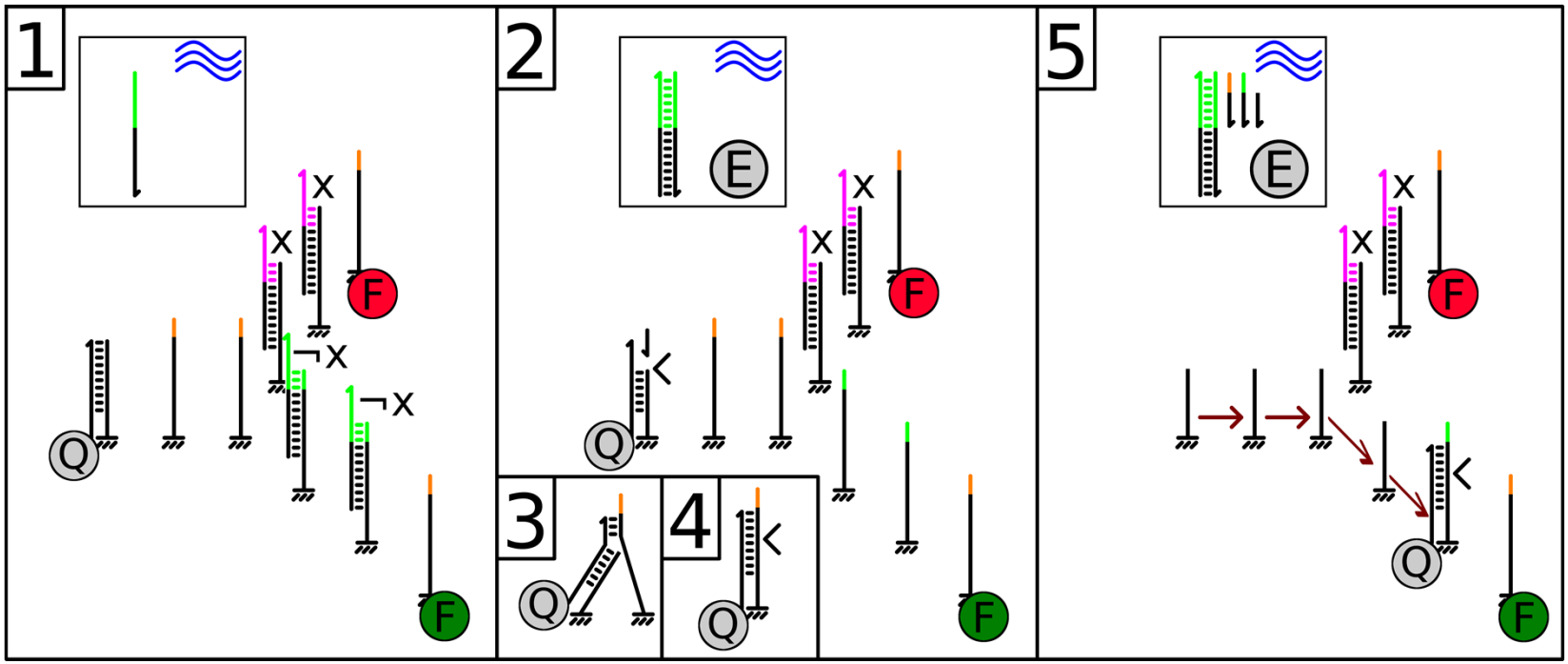
# Case study 4: DNA walkers

- How it works…
  - tracks made up of anchor strands laid out on DNA origami tile
  - can make molecule 'walk' by attaching/ detaching from anchor
  - autonomous, constant average speed
  - can control movement
  - can carry cargo
  - all made from DNA




Direct observation of stepwise movement of a synthetic molecular transporter. Wickham *et al*, Nature Nanotechnology 6, 166–169 (2011)

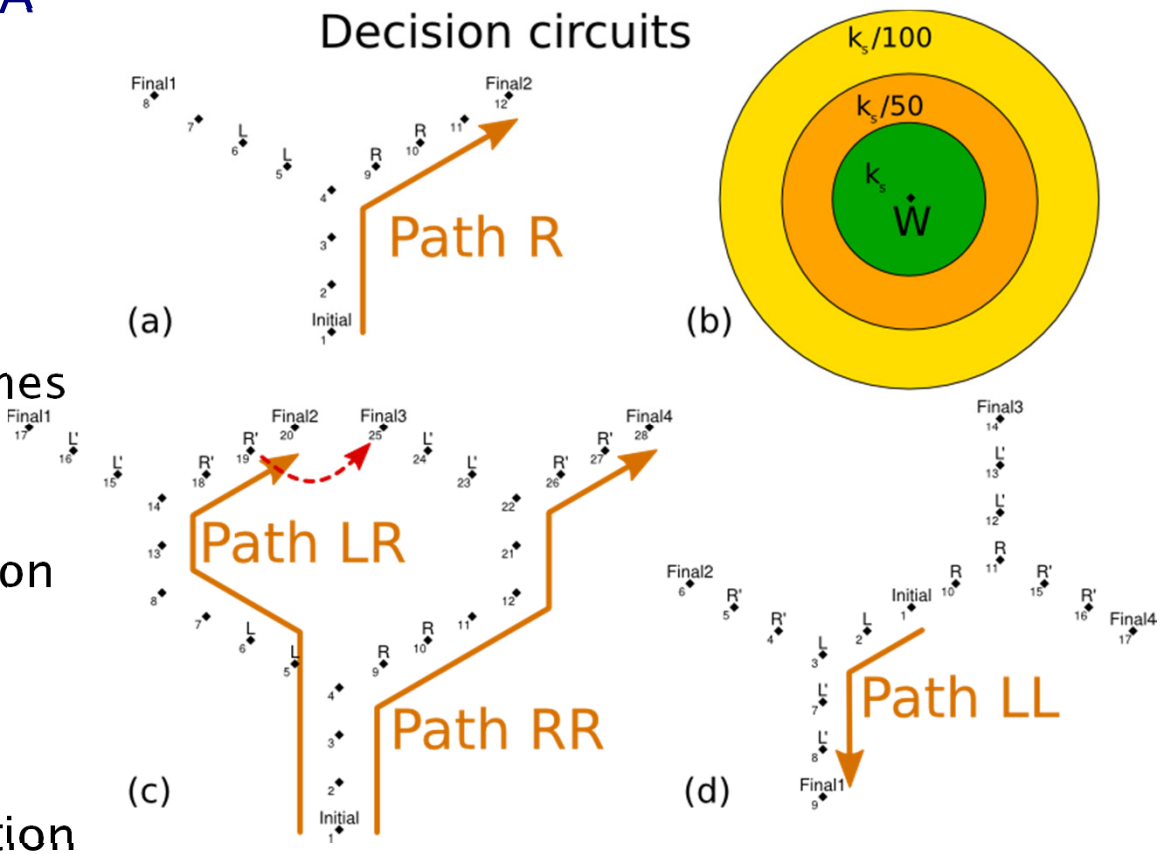
# Walker stepping action in detail…




1. Walker carries a quencher (Q)
2. Sections of the track can be selectively unblocked
3. Walker detaches from anchor strand
4. Walker attaches to the next anchor along the track
5. Fluorophores (F) detect walker reaching the end of the track

# DNA walker circuits

- Computing with DNA walkers
  - branching tracks laid out on DNA origami tile
  - starts at ‘initial’, signals when reaches ‘final’
  - can control ‘left’/’right’ decision
  - (this technology) single use only, ‘burns’ anchors
  - any Boolean function

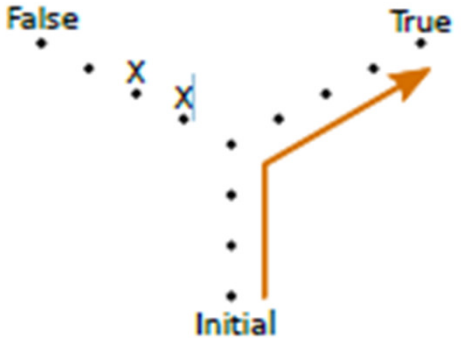


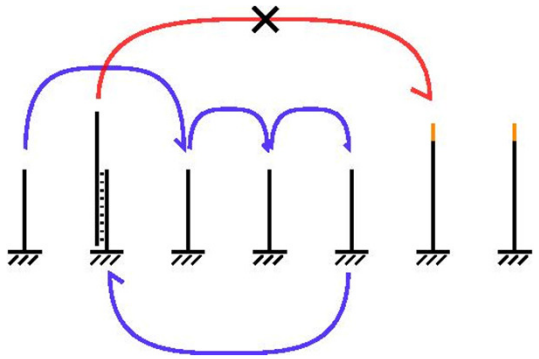

- Localised computation, well mixed assumption as in solution does not apply

DNA walker circuits: Computational potential, design, and verification. Dannenberg *et al*, Natural Computing, To appear, 2014

# DNA walkers: applications

- Walkers can realise biosensors: safety/reliability paramount
- Molecular walker computation inherently unreliable…
  - 87% follow the correct path
  - can jump over one or two anchorages, can deadlock

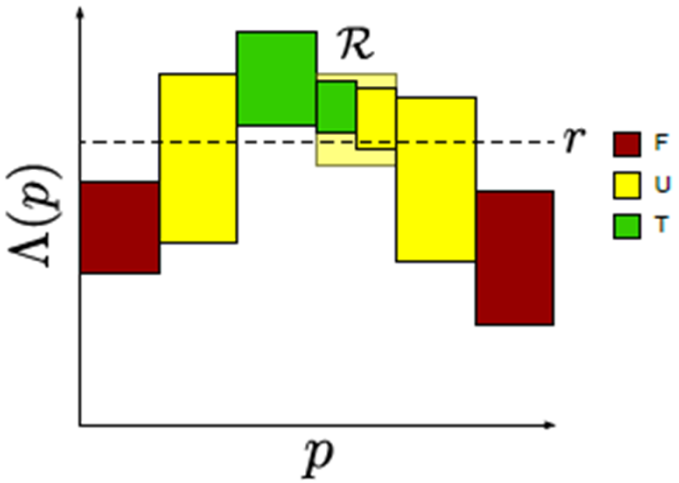


- Analyse reliability of molecular walker circuits using PRISM
  - devise a CTMC model, fit to experimental data
  - analyse reliability, deadlock and performance
  - use model checking results to improve the layout

# From verification to synthesis…

- Automated verification aims to establish if a property holds for a given model
- Can we find a model so that a property is satisfied?
  - difficult…
- The parameter synthesis problem is
  - given a parametric model, property and probability threshold
  - find a partition of the parameter space into True, False and Uncertain regions s.t. the relative volume of Uncertain is less or equal than a given ϵ
- Successive region refinement, based on over & under approx., implemented in PRISM

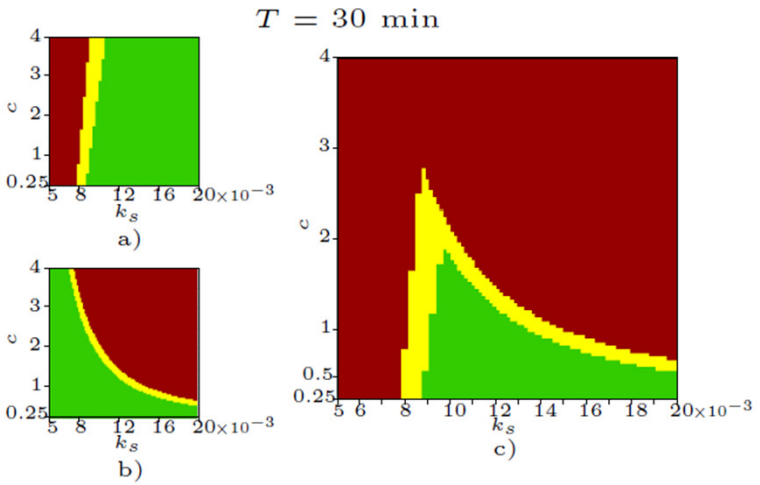



Precise Parameter Synthesis for Stochastic Biochemical Systems, Ceska *et al*, Proc. CMSB, To appear, 2014

# DNA walkers: parameter synthesis

- Application to biosensor design: can we synthesise the values of rates to guarantee a specified reliability level?
- For the walker model:
  - walker stepping rate k = funct ($k_s$, c,) where $k_s$ lies in interval [0.005,0.020], c in [0.25, 4]
  - find regions of values of $k_s$; and c where property is satisfied

a) $\Phi_1 = P_{\geq 0.4}[F^{[30,30]}$ finish-correct$]$
b) $\Phi_2 = P_{\leq 0.08}[F^{[30,30]}$ finish-incorrect$]$
c) $\Phi_1 \wedge \Phi_2$



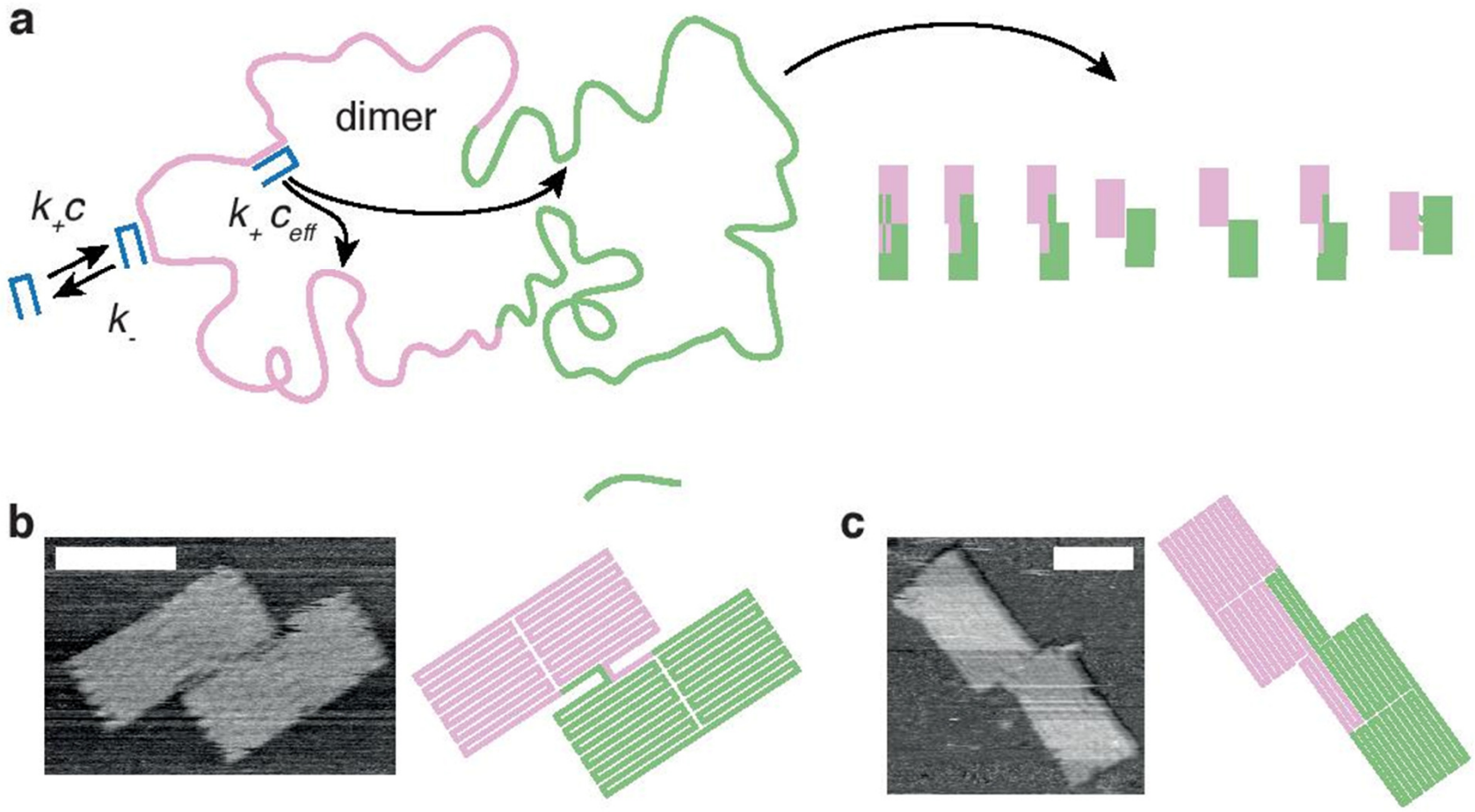

- Fast: for T=200, 88s with sampling, 329 subspaces

# Case study 5: Modelling DNA origami

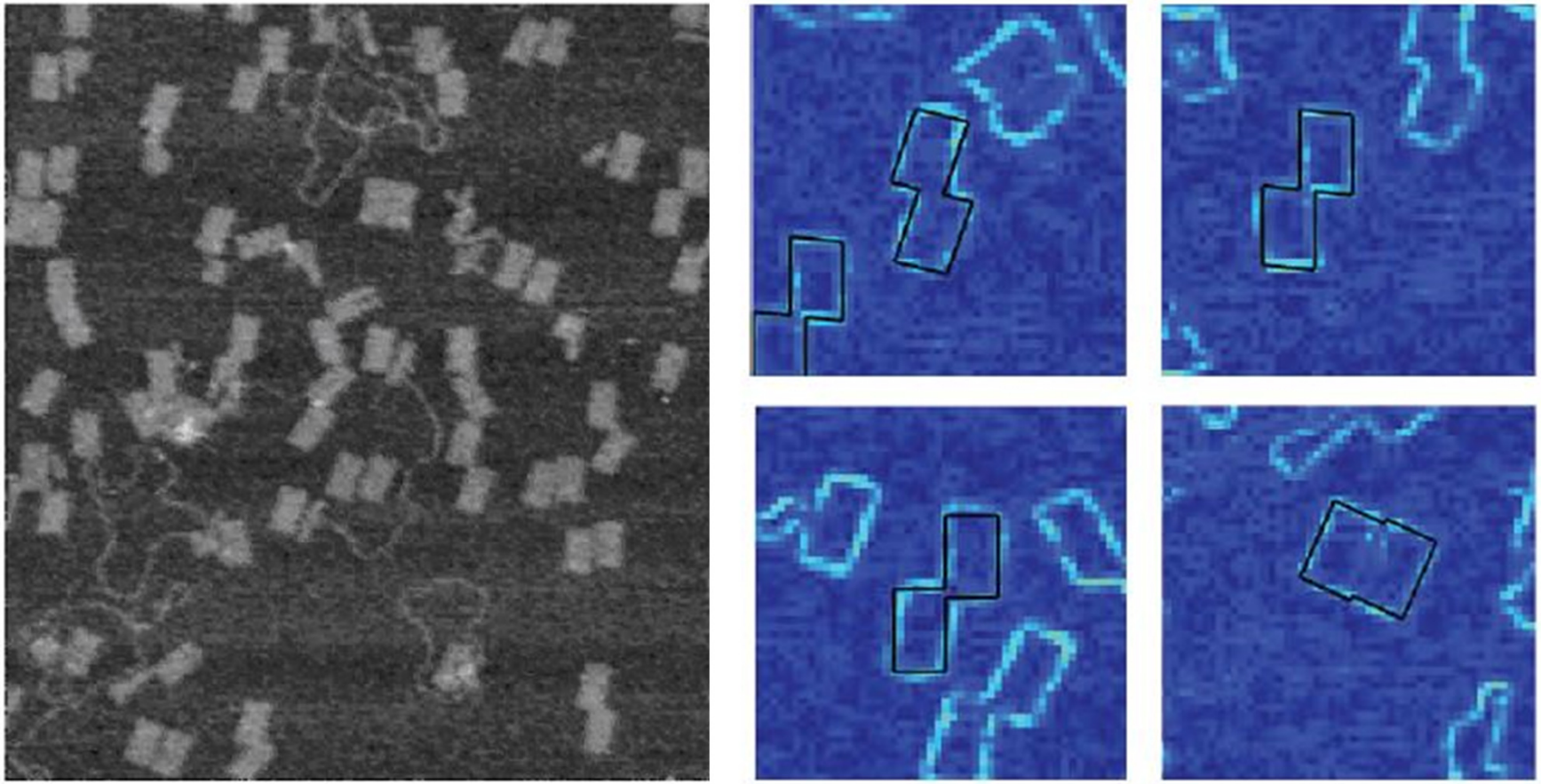
- DNA origami robust technique
  - robust assembly technique
  - folds into the single most stable shape
- Aim to understand how to control the folding pathways
  - develop a 'dimer' origami design, which has several well-folded shapes (planar and unstrained) corresponding to energy minima
  - formulate an abstract Markov chain model that is thermodynamically self-consistent
  - obtain model predictions using Gillespie simulation
  - perform a range of experiments (e.g. removing or cutting staples in half) that favour certain well-folded shapes
- Remarkably, the model is consistent with experimental observations

Guiding the folding pathway of DNA origami. Dunne, Dannenberg, Ouldridge, Kwiatkowska, Turberfield & Bath, Nature (in press)

# Dimer origami



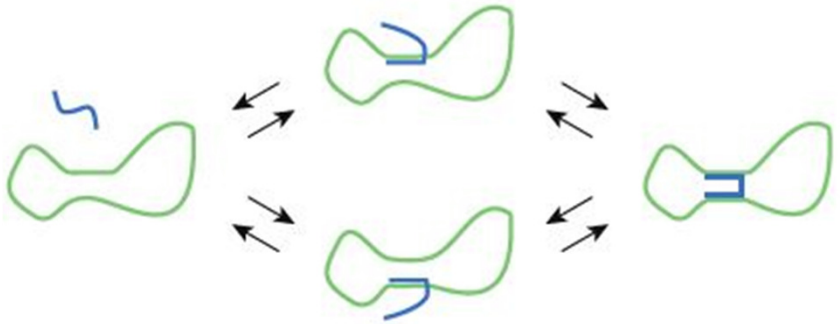
# Dimer shapes



- Develop image processing software to classify shapes

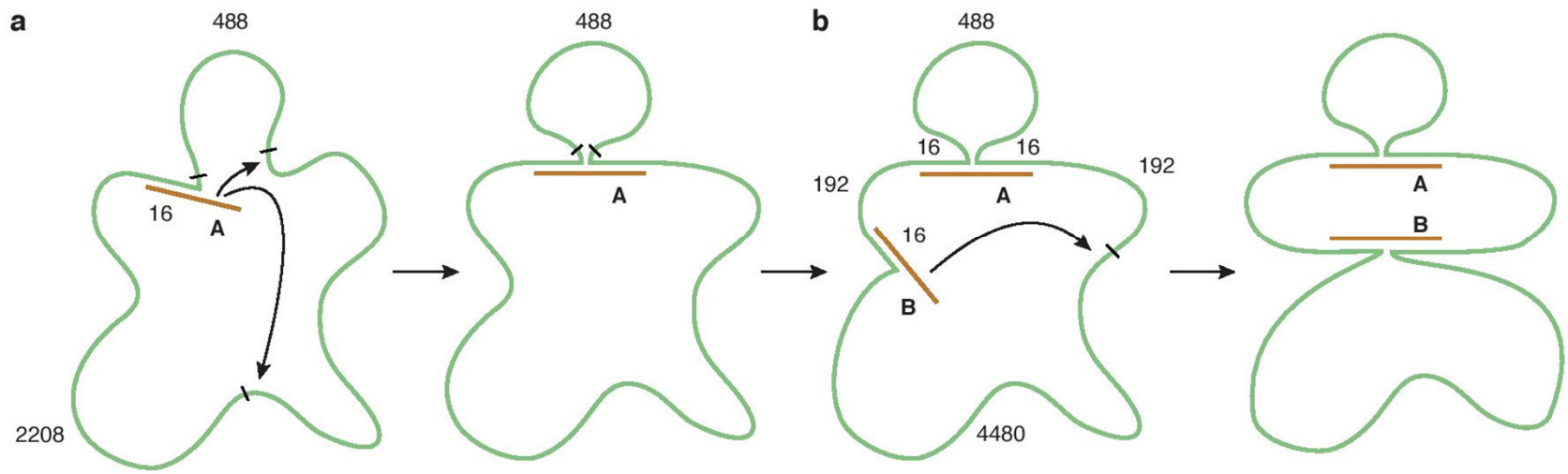
# The CTMC model

- Abstract the scaffold as a sequence of domains (16nt)
  - each staple has 2 positions to bind to
  - single-domain and two-domain staples
- State space
  - for monomer, 5 possibilities for two-domain staples
  - for dimer, $4^N \times 34^M$ ,
    N = 24 one-domain and
    M = 156 two-domain staples



- Rates (inhomogeneous CTMC)
  - can use mass action only for staple binding from solution
  - otherwise, estimate free energy change
  - need to consider loop formation…

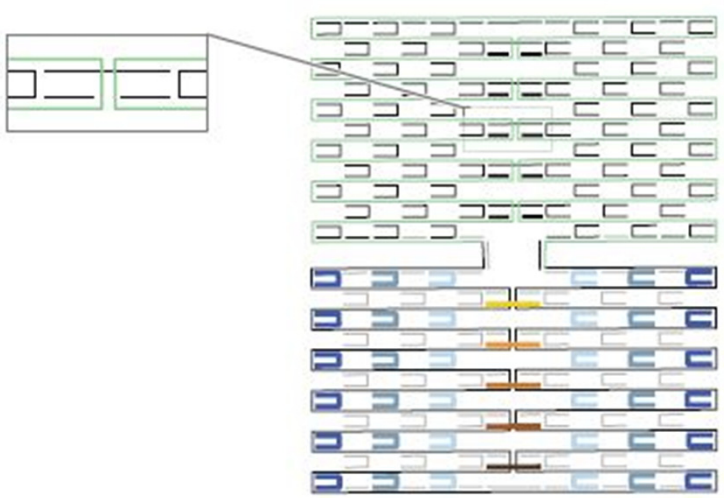
# Loop formation

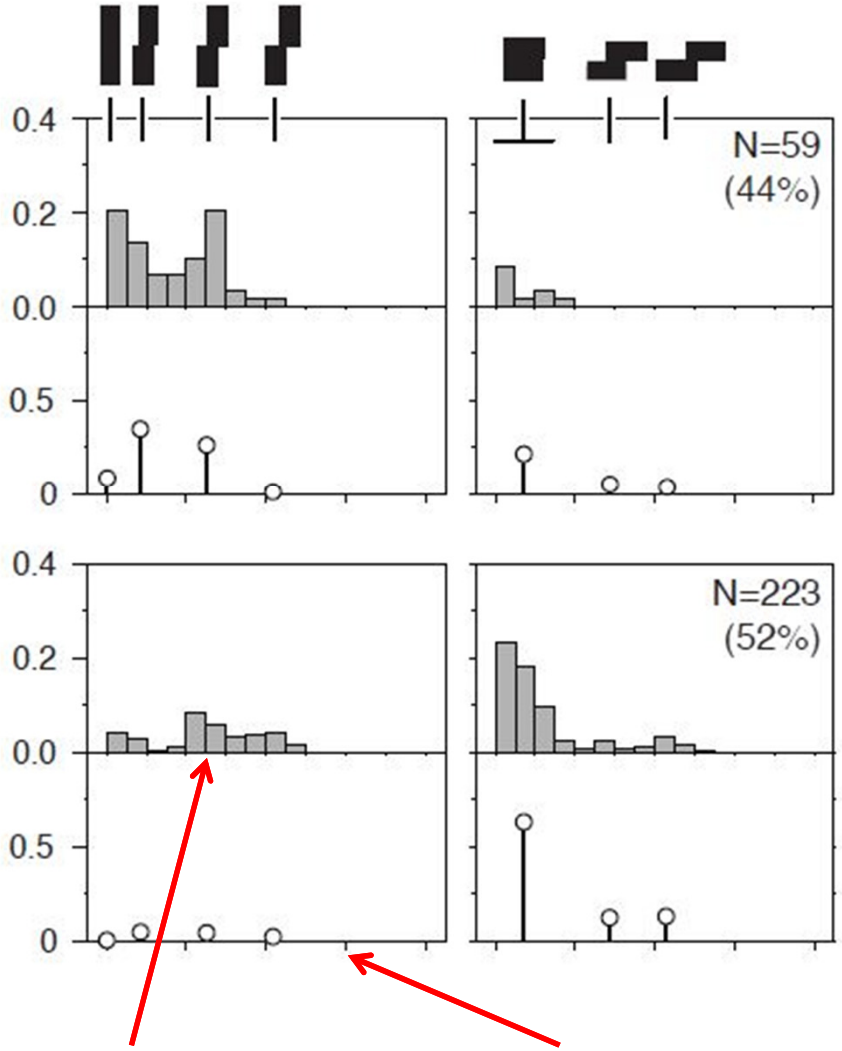



- Main idea: shortening of the loop by staple binding increases stability
  - use Dijkstra's shortest path algorithm to calculate adjustment in free energy
- Thus presence of staple A accelerates hybridization of B
- Planarity constraints

# Results on folding

- Distribution of shapes classified via offset
- Gillespie simulation






Modified tile

Observed shape

Predicted

# What has been achieved?

- Some successes
  - automatically found a flaw in DNA program
  - design automation for DNA walker circuits, can guarantee reliability levels, fast
  - improved scientific understanding of DNA origami folding
- Also failures: limited scalability (but see [CMSB 2015])
  - DNA transducer: 6–7 molecules
  - DNA walker circuits: smaller models can be handled with fast adaptive unformisation, lager ones only with statistical model checking, sometimes with better accuracy
  - DNA origami folding: only simulation is feasible
- Challenges
  - need to incorporate physics (thermodynamics, entropy, energy)

# Conclusions

- Demonstrated that quantitative/probabilistic verification can play a central role not only in systems biology, but also in design automation of molecular devices

- Many positive results:
  - predictive models
  - successful experimental validation
  - demonstrated practical feasibility of probabilistic modelling and verification in some contexts

- Key challenge (as always): state space explosion
  - can we exploit compositionality in analysis?
  - can we synthesise walker circuit layout? origami designs?
  - parameter/model synthesis for more complex models…

# Acknowledgements


- My group and collaborators in this work
- Project funding
  - ERC, EPSRC, Microsoft Research
  - Oxford Martin School, Institute for the Future of Computing


- See also
  - VERIWARE www.veriware.org
  - PRISM www.prismmodelchecker.org