Electronic Supplementary Material (ESI) for RSC Advances.

SUPPORTING INFORMATION FOR:

Renewable DNA Seesaw Logic Circuits Enabled by Photoregulation of Toehold-Mediated Strand Displacement

Xin Song[1,], Abeer Eshra[2,3], Chris Dwyer[1,2] and John Reif[1,2]*

[1] Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA

[2] Department of Computer Science, Duke University, Durham, NC, USA

[3] Department of Computer Science and Engineering, Faculty of Electronic Eng., Menoufia University, Menouf, Egypt

* Email: xin.song@duke.edu

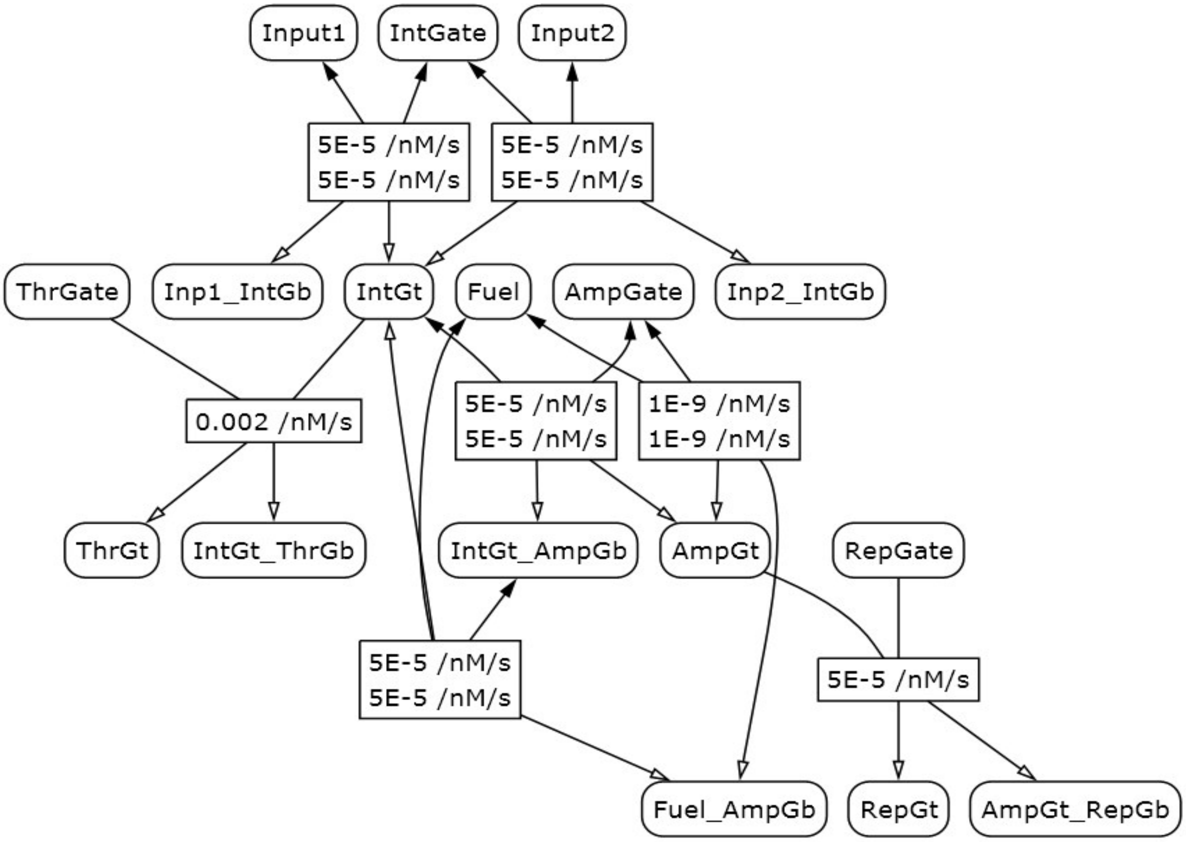

Figure S1. Reaction graph of a DNA seesaw logic AND gate modified with trans-form azobenzenes. Generated by Visual GEC.[1]

Table S1. Naming convention for reactants and products in the modeling

Species name	Meaning
Input1	ssDNA **input signal 1**
Input2	ssDNA **input signal 2**
IntGate	dsDNA integrating gate complex
ThrGate	dsDNA threshold gate complex
AmpGate	dsDNA amplifying gate complex
Fuel	ssDNA fuel strand
RepGate	dsDNA reporter gate complex
Inp1_IntGb	dsDNA complex formed by input signal 1 + integrating gate base strand
Inp2_IntGb	dsDNA complex formed by input signal 2 + integrating gate base strand
IntGt_ThrGb	dsDNA complex formed by integrating gate top strand + threshold gate base strand
IntGt_AmpGb	dsDNA complex formed by integrating gate top strand + amplifying gate base strand
Fuel_AmpGb	dsDNA complex formed by fuel strand + amplifying gate base strand
AmpGt_RepGb	dsDNA complex formed by amplifying gate top strand + reporter gate base strand
IntGt	ssDNA integrating gate top strand
ThrGt	ssDNA threshold gate top strand
AmpGt	ssDNA amplifying gate top strand
RepGt	ssDNA reporter gate top strand **(regarded as gate output signal in our model)**

Table S2. BM rate constants for modeling DNA seesaw logic AND gate with trans-form azobenzenes

BM rate constant	Rate constant value (nM^{-1}s^{-1})
ks	5e-05
kf	2e-03
kl	1e-09

Table S3. DNA species involved in the initial state of a seesaw logic AND gate

Species name	Initial concentration (nM)
Input1	90.0
Input2	90.0
IntGate	200.0
ThrGate	120.0
AmpGate	100.0
Fuel	200.0
RepGate	150.0

Table S4. Final state of seesaw AND gate forward operation under different Boolean input combinations

DNA species	Final conc. (nM) Input: ON-ON	Final conc. (nM) Input: ON-OFF	Final conc. (nM) Input: OFF-ON	Final conc. (nM) Input: OFF-OFF
Input1	17.7	0.1	0	0
Input2	17.7	0	0.1	0
IntGate	55.5	100.1	100.1	180
ThrGate	0	20.1	20.1	100
AmpGate	0.3	95.6	95.6	99.5
Fuel	111.2	195.6	195.6	199.5
RepGate	50.4	145.6	145.6	149.5
Inp1_IntGb	72.3	89.9	10	10
Inp2_IntGb	72.3	10	89.9	10
IntGt_ThrGb	120	99.9	99.9	20
IntGt_AmpGb	10.9	0	0	0
Fuel_AmpGb	88.8	4.4	4.4	0.5
AmpGt_RepGb	99.6	4.4	4.4	0.5
IntGt	13.6	0	0	0
ThrGt	120	99.9	99.9	20
AmpGt	0.1	0	0	0
RepGt	99.6 (Output ON)	4.4 (Output OFF)	4.4 (Output OFF)	0.5 (Output OFF)

Table S5. Estimation of ksc and the corresponding critical concentration

Invading toehold length n (nt)	Invading toehold binding energy (kcal/mol)^2	MATLAB command	MB rate constant ksc = $k_{\{5,n\}}$ (nM^{-1}s^{-1})	Critical concentration (nM)
0	+1.9	BM_rate([1.9,-6.7,15,25,3])	4.1410e-10	3.2839e+08
1	+0.2	BM_rate([0.2,-6.7,15,25,3])	7.2793e-09	1.8681e+07
2	-1.7	BM_rate([-1.7,-6.7,15,25,3])	1.7928e-07	7.5853e+05
3	-3.0	BM_rate([-3.0,-6.7,15,25,3])	1.6047e-06	8.4743e+04
4	-4.7	BM_rate([-4.7,-6.7,15,25,3])	2.7996e-05	4.8574e+03
5	-6.9	BM_rate([-6.9,-6.7,15,25,3])	8.6716e-04	156.8197

Table S6. Estimation of krc and the corresponding critical concentration

Incumbent toehold length m (nt)	Incumbent toehold binding energy (kcal/mol)[2]	MATLAB command	MB rate constant krc = $k_{\{m,5\}}$ (nM^{-1}s^{-1})	Critical concentration (nM)
0	+1.2	BM_rate([-6.9,1.2,15,25,3])	0.0011	168.7599
1	-0.6	BM_rate([-6.9,-0.6,15,25,3])	0.0011	168.7594
2	-2.7	BM_rate([-6.9,-2.7,15,25,3])	0.0011	168.7415
3	-4.5	BM_rate([-6.9,-4.5,15,25,3])	0.0010	168.3806
4	-5.6	BM_rate([-6.9,-5.6,15,25,3])	0.0010	166.4296
5	-6.7	BM_rate([-6.9,-6.7,15,25,3])	8.6716e-04	156.8197

Table S7. Estimation of tksc and the corresponding critical concentration

Temporary incumbent toehold length m (nt)	Invading toehold length n (nt) $n = 10 * \left(1 - \frac{m}{5}\right)$	Effective length of branch migration domain b (nt) $b = 15 - m$	MATLAB command	MB rate constant tksc = $k_{\{m,n\}}$ (nM^{-1}s^{-1})	Critical concentration (nM)
5	0	10	BM_rate([1.9,-6.7,10,25,3])	5.5582e-10	4.9259e+08
4	2	11	BM_rate([-1.7,-5.6,11,25,3])	2.9992e-07	1.0344e+06
3	4	12	BM_rate([-4.7,-4.5,12,25,3])	4.5100e-05	6.1018e+03
2	6	13	BM_rate([-8.3,-2.7,13,25,3])	0.0029	80.4382
1	8	14	BM_rate([-11.9,-0.6,14,25,3])	0.0035	58.3359
0	10	15	BM_rate([-14.8,1.2,15,25,3])	0.0035	50.7938

Table S8. Estimation of tkrc and the corresponding critical concentration

Invading toehold length n (nt)	Incumbent toehold length m (nt)	Effective length of branch migration domain b (nt) $b = 15 - n$	MATLAB command	MB rate constant tkrc = $k_{\{m,n\}}$ (nM^{-1}s^{-1})	Critical concentration (nM)
5	0	10	BM_rate([-6.9,1.2,10,25,3])	0.0014	291.2350
4	2	11	BM_rate([-4.7,-2.7,11,25,3])	4.9574e-05	6.6652e+03
3	4	12	BM_rate([-3,-5.6,12,25,3])	2.4730e-06	1.0595e+05
2	6	13	BM_rate([-1.7,-9.5,13,25,3])	6.6223e-09	8.7519e+05

Table S9. Estimation of rpksc and the corresponding critical concentration

Invading toehold length n (nt)	Incumbent toehold length m (nt) $m = 5 - n$	Effective length of branch migration domain b (nt) $b = 15 - m$	MATLAB command	MB rate constant rpksc = $k_{\{m,n\}}$ (nM^{-1}s^{-1})	Critical concentration (nM)
0	5	10	BM_rate([1.9,-6.7,10,25,3])	5.5582e-10	4.9259e+08
1	4	11	BM_rate([0.2,-5.6,11,25,3])	1.2178e-08	2.5475e+07
2	3	12	BM_rate([-1.7,-4.5,12,25,3])	2.9022e-07	9.4820e+05
3	2	13	BM_rate([-3,-2.7,13,25,3])	2.4190e-06	9.7803e+04
4	1	14	BM_rate([-4.7,-0.6,14,25,3])	3.9088e-05	5.2210e+03
5	0	15	BM_rate([-6.9,1.2,15,25,3])	0.0011	168.7599

Table S10. Estimation of rpkrc and the corresponding critical concentration

Invading toehold length n (nt)	Incumbent toehold length m (nt) $m = 5 - n$	Effective length of branch migration domain b (nt) $b = 15 - n$	MATLAB command	MB rate constant rpkrc = $k_{\{m,n\}}$ ($nM^{-1}s^{-1}$)	Critical concentration (nM)
5	0	10	BM_rate([-6.9,1.2,10,25,3])	0.0014	291.2350
4	1	11	BM_rate([-4.7,-0.6,11,25,3])	4.9597e-05	6.6652e+03
3	2	12	BM_rate([-3,-2.7,12,25,3])	2.6204e-06	1.0596e+05
2	3	13	BM_rate([-1.7,-4.5,13,25,3])	2.6809e-07	8.7525e+05
1	4	14	BM_rate([0.2,-5.6,14,25,3])	9.6965e-09	2.0016e+07
0	5	15	BM_rate([1.9,-6.7,15,25,3])	4.1410e-10	3.2839e+08

Table S11. Estimation of ampksc and the corresponding critical concentration

Invading toehold length n (nt)	Effective length of branch migration domain b (nt)	Incumbent toehold length m (nt) $m = 20 - b$	MATLAB command	MB rate constant ampksc = $k_{\{m,n\}}$ ($nM^{-1}s^{-1}$)	Critical concentration (nM)
0	10	10	N/A	N/A	N/A
1	11	9	N/A	N/A	N/A
2	12	8	N/A	N/A	N/A
3	13	7	BM_rate([-3,-10.2,13,25,3])	1.8529e-08	9.7736e+04
4	14	6	BM_rate([-4.7,-9.5,14,25,3])	1.0402e-06	5.1643e+03
5	15	5	BM_rate([-6.9,-6.7,15,25,3])	8.6716e-04	156.8197

Table S12. Estimation of ampkrc and the corresponding critical concentration

Invading toehold length n (nt) $n = 20 - b$	Effective length of branch migration domain b (nt)	Incumbent toehold length m (nt)	MATLAB command	MB rate constant ampkrc = $k_{\{m,n\}}$ ($nM^{-1}s^{-1}$)	Critical concentration (nM)
10	10	0	BM_rate([-14.8,1.2,10,25,3])	0.0035	114.2859
9	11	1	BM_rate([-12.9,-0.6,11,25,3])	0.0035	94.4561
8	12	2	BM_rate([-11.9,-2.7,12,25,3])	0.0035	79.3614
7	13	3	BM_rate([-9.2,-4.5,13,25,3])	0.0034	69.8579
6	14	4	BM_rate([-8.3,-5.6,14,25,3])	0.0029	67.3766
5	15	5	BM_rate([-6.9,-6.7,15,25,3])	8.6716e-04	156.8197

Table S13. Estimation of fksc and the corresponding critical concentration

Invading toehold length n (nt)	Effective length of branch migration domain b (nt)	Incumbent toehold length m (nt)	MATLAB command	MB rate constant fksc = $k_{\{m,5\}}$ ($nM^{-1}s^{-1}$)	Critical concentration (nM)
5	10	0	BM_rate([-6.9,1.2,10,25,3])	0.0014	291.2350
5	11	1	BM_rate([-6.9,-0.6,11,25,3])	0.0013	255.3127
5	12	2	BM_rate([-6.9,-2.7,12,25,3])	0.0012	226.7870
5	13	3	BM_rate([-6.9,-4.5,13,25,3])	0.0012	203.1576
5	14	4	BM_rate([-6.9,-5.6,14,25,3])	0.0011	181.8446
5	15	5	BM_rate([-6.9,-6.7,15,25,3])	8.6716e-04	156.8197

Table S14. Estimation of fkrc and the corresponding critical concentration

Invading toehold length n (nt)	**Effective length of branch migration domain b (nt)**	**Incumbent toehold length m (nt)**	**MATLAB command**	**MB rate constant fkrc = $k_{\{5,n\}}$ ($nM^{-1}s^{-1}$)**	**Critical concentration (nM)**
0	10	5	BM_rate([1.9,-6.7,10,25,3])	5.5582e-10	4.9259e+08
1	11	5	BM_rate([0.2,-6.7,11,25,3])	9.1446e-09	2.5475e+07
2	12	5	BM_rate([-1.7,-6.7,12,25,3])	2.1165e-07	9.4817e+05
3	13	5	BM_rate([-3,-6.7,13,25,3])	1.7869e-06	9.7785e+04
4	14	5	BM_rate([-4.7,-6.7,14,25,3])	2.9487e-05	5.2066e+03
5	15	5	BM_rate([-6.9,-6.7,15,25,3])	8.6716e-04	156.8197

Table S15. Estimation of ampfkrc and the corresponding critical concentration

Invading toehold length n (nt)	**Effective length of branch migration domain b (nt)**	**Incumbent toehold length m (nt)**	**MATLAB command**	**MB rate constant ampfkrc = $k_{\{0,n\}}$ ($nM^{-1}s^{-1}$)**	**Critical concentration (nM)**
5	15	0	BM_rate([-6.9,1.2,15,25,3])	0.0011	168.7599
4	15	0	BM_rate([-4.7,1.2,15,25,3])	3.6510e-05	4.8693e+03
3	15	0	BM_rate([-3,1.2,15,25,3])	2.0976e-06	8.4755e+04
2	15	0	BM_rate([-1.7,1.2,15,25,3])	2.3437e-07	7.5854e+05
1	15	0	BM_rate([0.2,1.2,15,25,3])	9.5163e-09	1.8681e+07
0	15	0	BM_rate([1.9,1.2,15,25,3])	5.4136e-10	3.2839e+08

Text S1: ODE Modeling of the Mass-Action Kinetics of Seesaw Logic AND Gate Forward Operation. Generated by Dynetica.[3]

```
d[Input1]/dt = -1.0 * (ks * [Input1] * [IntGate]) + ks * [Inp1_IntGb] * [IntGt]
d[Input2]/dt = -1.0 * (ks * [Input2] * [IntGate]) + ks * [Inp2_IntGb] * [IntGt]
d[IntGate]/dt = (-1.0 * (ks * [Input1] * [IntGate]) + ks * [Inp1_IntGb] * [IntGt]) - ks * [Input2] * [IntGate] + ks * [Inp2_IntGb] * [IntGt]
d[ThrGate]/dt = -1.0 * (kf * [IntGt] * [ThrGate])
d[AmpGate]/dt = (-1.0 * (ks * [IntGt] * [AmpGate]) + ks * [IntGt_AmpGb] * [AmpGt]) - kl * [AmpGate] * [Fuel] + kl * [Fuel_AmpGb] * [AmpGt]
d[Fuel]/dt = (-1.0 * (ks * [IntGt_AmpGb] * [Fuel]) + ks * [Fuel_AmpGb] * [IntGt]) - kl * [AmpGate] * [Fuel] + kl * [Fuel_AmpGb] * [AmpGt]
d[RepGate]/dt = -1.0 * (ks * [AmpGt] * [RepGate])
d[Inp1_IntGb]/dt = ks * [Input1] * [IntGate] - ks * [Inp1_IntGb] * [IntGt]
d[Inp2_IntGb]/dt = ks * [Input2] * [IntGate] - ks * [Inp2_IntGb] * [IntGt]
d[IntGt_ThrGb]/dt = kf * [IntGt] * [ThrGate]
d[IntGt_AmpGb]/dt = ks * [IntGt] * [AmpGate] - ks * [IntGt_AmpGb] * [AmpGt] - ks * [IntGt_AmpGb] * [Fuel] + ks * [Fuel_AmpGb] * [IntGt]
d[Fuel_AmpGb]/dt = (ks * [IntGt_AmpGb] * [Fuel] - ks * [Fuel_AmpGb] * [IntGt] + kl * [AmpGate] * [Fuel]) - kl * [Fuel_AmpGb] * [AmpGt]
d[AmpGt_RepGb]/dt = ks * [AmpGt] * [RepGate]
d[IntGt]/dt = (((ks * [Input1] * [IntGate] - ks * [Inp1_IntGb] * [IntGt] + ks * [Input2] * [IntGate]) - ks * [Inp2_IntGb] * [IntGt] - ks * [IntGt] * [AmpGate] + ks * [IntGt_AmpGb] * [AmpGt]) - kf * [IntGt] * [ThrGate] + ks * [IntGt_AmpGb] * [Fuel]) - ks * [Fuel_AmpGb] * [IntGt]
d[ThrGt]/dt = kf * [IntGt] * [ThrGate]
```

```
d[AmpGt]/dt = (ks * [IntGt] * [AmpGate] - ks * [IntGt_AmpGb] * [AmpGt] - ks * [AmpGt] * [RepGate] + kl * [AmpGate] * [Fuel]) -
kl * [Fuel_AmpGb] * [AmpGt]
d[RepGt]/dt = ks * [AmpGt] * [RepGate]

Parameter values:
-----------------------------------
ks = 0.00005
kf = 0.002
kl = 0.000000001
```

Text S2: MATLAB Code (with Modifications on Units of Numerical Values) for Estimating BM Rate Constants Associated with Seesaw Gate Renewal Process. Complete MATLAB script was adapted from the Supporting Information of Prior Publication by Zhang and Winfree.[2]

```
%BM rate constant (in unit of nM-1s-1)
BM_rate = 1E-9 * kf * kb * incumbent_offrate / (invading_offrate * incumbent_offrate
...
+ kb * invading_offrate + kb * incumbent_offrate)

%critical concentration (in unit of nM)
c_crit = 1E9 * (0.1 / kf) * (invading_offrate * incumbent_offrate + kb *
invading_offrate ...
+ kb * incumbent_offrate) / (incumbent_offrate + kb)

output = [BM_rate, c_crit];
```

Text S3: LBS Program for Simulating the Mass-Action Kinetics of a Seesaw Logic AND Gate Functionalized with Cis-Form Azobenzenes.

```
//////////////////////////////////////
// LBS code for seesaw gate renewal process
// Gate type: AND
// Input combination: ON, ON
// Effective azobenzene isomerization yield: 80%
//////////////////////////////////////

directive sample 36000.0 2000

rate ksc = 7.2793e-09;
rate krc = 0.0011;
rate tksc = 2.9992e-07;
rate tkrc = 4.9574e-05;
rate rpksc = 1.2178e-08;
rate rpkrc = 4.9597e-05;
rate ampksc = 0;
rate ampkrc = 0.0035;
rate fksc = 0.0013;
rate fkrc = 9.1446e-09;
rate ampfksc = 0;
rate ampfkrc = 3.6510e-05;

spec Input1 = new Input1;
```

```
spec Input2 = new Input2;
spec RepGt = new RepGt;

init Input1 17.7 |
init Input2 17.7 |

module gateAND(spec Input1, Input2){

        spec IntGate = new IntGate;
        spec ThrGate = new ThrGate;
        spec AmpGate = new AmpGate;
        spec Fuel = new Fuel;
        spec RepGate = new RepGate;
        spec Inp1_IntGb = new Inp1_IntGb;
        spec Inp2_IntGb = new Inp2_IntGb;
        spec IntGt_ThrGb = new IntGt_ThrGb;
        spec IntGt_AmpGb = new IntGt_AmpGb;
        spec Fuel_AmpGb = new Fuel_AmpGb;
        spec AmpGt_RepGb = new AmpGt_RepGb;
        spec IntGt = new IntGt;
        spec ThrGt = new ThrGt;
        spec AmpGt = new AmpGt;

        init IntGate        55.5    |       // integrating gate
        init ThrGate        0       |       // threshold gate
        init AmpGate        0.3     |       // amplifying gate
        init Fuel           111.2   |       // Fuel
        init RepGate        50.4    |       // reporter gate

        init Inp1_IntGb     72.3    |       // input 1 + integrating gate bottom
        init Inp2_IntGb     72.3    |       // input 2 + integrating gate bottom
        init IntGt_ThrGb    120     |       // integrating top + threshold gate bottom
        init IntGt_AmpGb    10.9    |       // integrating top + amplifying gate bottom
        init Fuel_AmpGb     88.8    |       // Fuel + amplfying gate bottom
        init AmpGt_RepGb 99.6       |       // amplifying gate top + reporter gate bottom

        init IntGt          13.6    |       // integrating gate top
        init ThrGt          120     |       // threshold gate top
        init AmpGt          0.1     |       // amplifying gate top
        init RepGt          99.6    |       // reporter gate top

 // regeneration of integrating gate and inputs
 Input1 + IntGate <->{ksc}{krc} Inp1_IntGb + IntGt                            |
 Input2 + IntGate <->{ksc}{krc} Inp2_IntGb + IntGt                            |

 // regeneration of threshold gate
 IntGt + ThrGate <->{tksc}{tkrc} IntGt_ThrGb + ThrGt                  |

 // regeneration of reporter gate
 AmpGt + RepGate <->{rpksc}{rpkrc} AmpGt_RepGb + RepGt                |

 // regeneration of amplifying gate and fuel
 IntGt + AmpGate <->{ampksc}{ampkrc} IntGt_AmpGb + AmpGt              |
 IntGt_AmpGb + Fuel <->{fksc}{fkrc} Fuel_AmpGb + IntGt                |
 Fuel + AmpGate <->{ampfksc}{ampfkrc} Fuel_AmpGb + AmpGt

};
```

```
gateAND(Input1, Input2)
```

REFERENCES

1 M. Pedersen and A. Phillips, *J. R. Soc. Interface R. Soc.*, 2009, **6**, S437–S450.
2 D. Y. Zhang and E. Winfree, *J. Am. Chem. Soc.*, 2009, **131**, 17303–17314.
3 L. You, A. Hoonlor and J. Yin, *Bioinformatics*, 2003, **19**, 435–436.